



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## FACULTAD DE INFORMÁTICA

# TESINA DE LICENCIATURA

**TÍTULO:** Diseño de interacciones vibrotáctiles en dispositivos móviles

**AUTORES:** Julian Agustin Pasquale

**DIRECTOR:** Andrés Rodríguez

**CODIRECTOR:** Javier Bazzocco

**ASESOR PROFESIONAL:**

**CARRERA:** Licenciatura en Sistemas

### Resumen

El presente trabajo de grado se ubica en el área del diseño de experiencias de usuario que incluyen estímulos hápticos. El objetivo de esta tesina es contribuir con los diseñadores de estas experiencias mediante herramientas que se integren en su proceso de trabajo y faciliten la asociación de efectos de vibración con los otros componentes de las interfaces de usuario en aplicaciones móviles.

### Palabras Clave

Háptica, Móvil, Vibración, Prototipado, Diseño de experiencias de usuario.

### Conclusiones

Mediante el desarrollo de una aplicación web que permite diseñar vibraciones y el uso de una herramienta de diseño y prototipado web, Framer, fue posible no solo el acoplamiento de vibraciones a diseños sino también la posibilidad de generar prototipos interactivos que pueden ser probados desde dispositivos móviles, generando vibraciones que permiten dar retroalimentación háptica a los usuarios del sistema a desarrollar.

### Trabajos Realizados

Se realizó una exhaustiva investigación sobre háptica y los mecanorreceptores de la piel, así como también los trabajos relacionados a este tema y herramientas existentes.

Se desarrolló un sistema web que permite a diseñadores crear efectos vibrotáctiles y también se investigaron las herramientas de prototipado más utilizadas por los diseñadores en la industria en búsqueda de una herramienta que permita probar las vibraciones diseñadas en los dispositivos móviles.

### Trabajos Futuros

Desarrollo de aplicaciones nativas (Android o iOS) que permitan la creación y prueba de vibraciones en el mismo dispositivo.

Herramienta que permita crear vibraciones en base a la interacción del usuario con el dispositivo estudiando su comportamiento, o en base a dibujos de patrones.

Fabricar un dispositivo de hardware que permita probar las vibraciones creadas, independiente de cualquier dispositivo móvil.

**Fecha de la presentación: Agosto 2021**

# Índice

<b>Capítulo 1: Introducción</b>	<b>3</b>
1.1 Motivación	3
1.2 Objetivos	4
1.3 Estructura de la tesina	4
<b>Capítulo 2: Background y trabajos relacionados</b>	<b>6</b>
2.1 Introducción	6
2.2 El tacto humano	7
2.3 Trabajos relacionados	10
2.4 Diseño de experiencia de usuario	18
2.5 Conclusión	32
<b>Capítulo 3: Propuesta</b>	<b>33</b>
3.1 Introducción	33
3.2 Escenario de uso	33
3.3 Selección de plataforma de base	34
3.4 Tecnologías para la interacción háptica en móviles	42
3.5 Arquitectura de la solución	52
3.6 Demostración	60
<b>Capítulo 4: Conclusiones y posibles trabajos futuros</b>	<b>84</b>
<b>Capítulo 5: Bibliografía</b>	<b>86</b>
<b>Capítulo 6: Anexos</b>	<b>91</b>

# Capítulo 1: Introducción

El presente trabajo de grado se ubica en el área del diseño de experiencias de usuario que incluyen estímulos hápticos. El objetivo de esta tesina es contribuir con los diseñadores de estas experiencias mediante herramientas que se integren en su proceso de trabajo y faciliten la asociación de efectos de vibración con los otros componentes de las interfaces de usuario en aplicaciones móviles.

## 1.1 Motivación

El diseño de experiencias hápticas (en particular mediante efectos vibrotáctiles) es una tendencia que está siendo cada vez más relevante dentro de la industria tecnológica, ya sea en el área de los videojuegos, dispositivos de IoT<sup>1</sup> o desarrollo de aplicaciones móviles. Sin embargo, diseñar este tipo de experiencias es por el momento un gran reto para los diseñadores por la falta de herramientas específicas [2].

Aunque la mayoría de los dispositivos móviles contienen el hardware adecuado para generar estímulos vibrotáctiles, aún existen pocas herramientas que faciliten el diseño o acoplamiento de vibraciones durante la interacción con la lógica propia del trabajo de los diseñadores. Los inconvenientes que suelen presentarse son en su mayoría, por ejemplo, un conjunto limitado de muestras de vibraciones, o la imposibilidad de personalización por parte del usuario de la herramienta, o una interfaz no muy simple de utilizar para personas ajenas al rubro del diseño o informática, lo que las hace no aptas para novatos.

Como consecuencia de la escasez de este tipo de herramientas, los diseñadores deben realizar muchas iteraciones sobre los prototipos que los desarrolladores les proveen, ya que es la única forma que tienen de realizar pruebas y comprobar si lo desarrollado coincide efectivamente con el efecto que ellos esperaban que incluya el prototipo.

Aunque existen avances en términos de facilidades para los diseñadores de háptica, aún hay una gran diferencia en comparación, por ejemplo, con los diseñadores de interfaces gráficas, que cuentan con un gran abanico de herramientas a su disposición y permiten incluso a los clientes una validación de los prototipos, previa a que los desarrolladores realicen la implementación, acelerando así los tiempos de entrega del software en cuestión.

Una de las particularidades de la percepción háptica es que las diferencias entre experiencias personales puede ser muy alta. Esto representa un gran problema a la hora de realizar un desarrollo para un cliente, ya que la única instancia existente para que éste vea los progresos durante el desarrollo son las demos con la presentación de prototipos. No existe actualmente una herramienta que permita a los diseñadores definir en conjunto con el cliente cuál es el efecto vibratorio durante la elicitación de requerimientos. Si bien el diseñador puede haber entendido el mensaje que el cliente quería que su vibración transmita, dada la subjetividad mencionada anteriormente, podrían no percibir lo mismo frente a la misma vibración. Por lo que es muy probable que haya que realizar

---

<sup>1</sup> Internet of Things. Término que se utiliza para hacer referencia a la interconexión de los elementos de la vida cotidiana, como por ejemplo heladeras, termostatos, lámparas, etc. [1]

modificaciones en el prototipo presentado, extendiendo los plazos de desarrollo y generando disconformidad.

Otro factor a tener en cuenta es que no existe un modelo conceptual unificado entre diseñadores y stakeholders<sup>2</sup>, lo que genera problemas de comunicación y dificulta el diseño de las experiencias. Como consecuencia se deja totalmente en mano de los usuarios finales del sistema la experiencia o sensación producida por las vibraciones.

Sumado a esto, es necesario tener en cuenta que los diversos sistemas operativos y configuraciones de hardware se comportan de manera diferente ante cada vibración, lo que dificulta aún más tener un modelo conceptual universal y ser consistentes en el desarrollo de efectos vibratorios. Según el dispositivo que utilice el usuario para acceder al sistema, éste podrá proveer estímulos de distinta intensidad o comportamiento.

## 1.2 Objetivos

Con esta tesina aspiramos a alcanzar los siguientes objetivos:

### Objetivo principal

Contribuir a la integración de interacciones hápticas en el diseño de la experiencia de usuario de aplicaciones móviles.

### Objetivos específicos

- Comprender el proceso de diseño de UX<sup>3</sup> de aplicaciones móviles.
- Relevar y analizar las tecnologías y herramientas disponibles para el diseño de interacciones hápticas.
- Proponer y desarrollar una herramienta de soporte a los diseñadores de UX para que puedan incorporar las interacciones hápticas.

## 1.3 Estructura de la tesina

El resto del documento se organiza de la siguiente manera:

**Capítulo 1:** Introducción. Planteo del tema central del trabajo, objetivos, alcance y organización de sus partes.

**Capítulo 2:** Background y trabajos relacionados. Contiene el marco teórico y los trabajos previos relacionados con el tema.

---

<sup>2</sup> Trabajadores, organizaciones sociales, inversores y todos aquellos actores interesados en una empresa o producto, y que pueden verse afectados por sus actividades.

<sup>3</sup> Experiencias de usuario, por sus siglas en inglés UX (user experience).

**Capítulo 3:** Propuesta. Describe la herramienta desarrollada, junto con sus decisiones y especificaciones técnicas y su validación mediante demostración.

**Capítulo 4:** Conclusiones y posibles trabajos futuros. Contiene la conclusión del trabajo en base al objetivo planteado, junto la presentación de posibles trabajos futuros.

**Capítulo 5:** Bibliografía. Contiene enlaces a documentos, libros y fuentes en las que este trabajo se basa.

**Capítulo 6:** Anexos. Contiene artículos independientes, que explican o esclarecen escenarios que el trabajo permite resolver.

# Capítulo 2: Background y trabajos relacionados

## 2.1 Introducción

El término háptica fue utilizado en el año 1892 por el investigador alemán Max Dessoir para referirse al estudio del tacto, similar a “óptico” para la vista o “acústico” para el sonido [3]. En la actualidad, se utiliza tanto para referirse al estudio de la psicología y percepción del sentido del tacto, como a la tecnología que utiliza el tacto como método para brindar retroalimentación o feedback.

La tecnología háptica generalmente se divide en dos clases según la modalidad de sentido principal: sensaciones táctiles (o cutáneas) y propiocepción (o el sentido de ubicación y fuerza del cuerpo), este último incluye sentidos cinestésicos de fuerza y movimiento. Estos dos tipos de feedback son usados para distintos propósitos, por ejemplo, las personas usan los sentidos táctiles de su dedo para derivar la textura, pero la retroalimentación cinestésica para inferir el peso [4]; diferentes sentidos pueden ser combinados para obtener resultados más convincentes [5].

El feedback háptico puede proveer múltiples beneficios en experiencias interactivas. Las tecnologías hápticas permiten la transferencia de información entre humanos y computadoras; esta transferencia es rica, proximal y rápida. La información fluye en ambos sentidos, a través de la entrada y la salida, a veces simultáneamente. En particular nos referimos a los efectos hápticos diseñados.

La tecnología háptica puede reforzar otras modalidades, enriqueciendo la retroalimentación para una experiencia más completa, o proporcionar retroalimentación complementaria, con muchas razones posibles: saturación de información, por ejemplo, cuando las pantallas visuales o de audio han maximizado su salida; contexto de la tarea, por ejemplo, cuando el usuario está conduciendo y debe mantener la vista en la carretera; situaciones de deterioro de alguno de los sentidos, por ejemplo, cuando un usuario tiene visión o audición limitadas; pantallas ambientales, por ejemplo, mantener al usuario al tanto de un dato sin interrumpirlo; o naturaleza de la información, por ejemplo, comunicar emoción. El tacto puede ser usado como subtítulo de otros sentidos [6], y el resultado puede ser dramático, por ejemplo, una persona ciega que usa una pantalla basada en la lengua para ver una bola rodando lo suficientemente bien como para golpearla mientras cae [7]. Múltiples dispositivos han sido desarrollados y estudiados para personas con deficiencias visuales [8], [9].

Por supuesto, el tacto es un sentido rico por derecho propio. El tacto puede ser invisible, como el sonido, y espacial, como la visión. Es el primer sentido que desarrollamos, jugando un papel importante en las experiencias formativas [10]; las acciones sensoriomotoras pueden ayudar a reforzar la comprensión a través del aprendizaje incorporado [11]. Sentir un objeto es especialmente útil para discernir las propiedades de los materiales [4]. El tacto también puede ser usado para expresiones artísticas: Gunther et al [12] estudió un traje vibrotáctil de cuerpo completo para crear "surcos cutáneos" parecidos a la música, ayudando a identificar el espacio artístico de las sensaciones de TV, incluidos

conciertos con composiciones táctiles. Si bien la retroalimentación háptica puede mejorar la usabilidad y el rendimiento de una tarea [13], [14], el tacto está especialmente conectado a conexiones emocionales viscerales. La investigación de mercados ha estudiado múltiples formas en que el tacto puede conectarse con los clientes: la forma en que se siente un teléfono inteligente puede influir en una compra sobre una alternativa que podría funcionar mejor, y los clientes prefieren comprar en tiendas que les permiten tocar productos [10], [15].

## 2.2 El tacto humano

### Percepción háptica activa

El tacto es una de las modalidades sensoriales de los seres vivos, y nos permite percibir cualidades o características particulares del medio y los objetos a través de la piel o el sistema músculo-esquelético [16].

Podríamos decir que se conoce como percepción háptica manual a la habilidad que tienen las personas de identificar las formas, texturas o peso de los elementos con los que interactúan, a través del movimiento de los dedos. También nos permite reconocer fácil y rápidamente objetos familiares, aún sin la ayuda de otros sentidos como la vista. Se enfoca en las propiedades externas de los objetos [17].

A diferencia de sentidos como la vista, donde es fácil determinar el color de un objeto, para determinar la textura o suavidad de un objeto debemos mover los dedos sobre su superficie, debemos usar el tacto de forma activa.

La mano es la estructura principal utilizada en la exploración táctil del mundo externo. A diferencia del tacto pasivo, la percepción activa se basa en que la persona realiza movimientos voluntarios para inspeccionar los objetos de su entorno, lo que permite obtener información mucho más detallada, que está estrictamente relacionada con los movimientos realizados por la persona en su interacción con el objeto.

La detección háptica es un sentido crítico para las personas y cómo éstas experimentan o se relacionan con el mundo, ya que nos brinda la información necesaria para saber la fuerza justa que necesitamos aplicar sobre un objeto para agarrarlo, o bien la presión necesaria que debemos utilizar para apretar una tecla.

Uno de los ejemplos más claros y con el que nos encontramos las personas a diario es cuando movemos nuestra mano a oscuras sobre la pared para identificar el interruptor de luz.

### Kinestesia

El sentido de háptica nos proporciona información a través de receptores que se encuentran en la piel (mecanoreceptores) y en los músculos, es decir que no incluye sólo al sentido del tacto sino que también el de kinestesia que se basa en el movimiento [18].

Es común encontrar referencias sobre la kinestesia junto a la propiocepción, que abarca el sentido de la postura y equilibrio con información de los tendones, músculos y

articulaciones. En particular la propiocepción se refiere a cómo las personas perciben su propio cuerpo, a diferencia del sentido háptico que es como las personas perciben a los objetos externos.

La información kinestésica proviene de sensores ubicados en los músculos, llamados husos musculares, que proveen al sistema nervioso información sobre la longitud y posición de los músculos. De esta forma el cuerpo es capaz de percibir la posición de los músculos, y la dirección, amplitud y velocidad de los movimientos, como así también enterarse ante cada cambio de estas propiedades.

Cuando una persona mueve alguno de sus músculos, la piel se estira o se contrae, y esa es la información que procesa el sistema nervioso para determinar los movimientos y la fuerza generada por los músculos que se mueven de manera voluntaria, a través del órgano tendinoso de Golgi, que se encuentra en la unión entre las fibras musculares y los filamentos de colágeno de los tendones.

Al mover los dedos se producen pequeñas y complejas vibraciones que son captadas por todo tipo de mecanorreceptores en la piel. La información obtenida por nuestros sentidos es transmitida al cerebro, cuyas capas la procesan y nos permiten obtener la información necesaria para nuestras experiencias hápticas.

## La piel y mecanorreceptores

Como se mencionó previamente, la información cinestésica es proporcionada por receptores de la piel, llamados mecanorreceptores [19].

Dentro de la piel se identifican 4 cualidades de sensibilidad, generalmente conocidas como sentido o sensibilidad cutánea. Éstas son la temperatura, presión o tacto, dolor y picazón:

- El tacto y la temperatura nos proveen información espacial y temporal de nuestro cuerpo. Gracias a estos receptores podemos encontrar por ejemplo una moneda en nuestro bolsillo, o saber si una taza se encuentra muy caliente.
- Los receptores de dolor protegen a la piel contra posibles daños y pueden responder ante estímulos térmicos, mecánicos o químicos. También se hace referencia a esto como nocicepción, y permite identificar distintos tipos de elementos que puedan dañar los tejidos, como el filo de un cuchillo.
- La picazón puede ser una sensación transitoria debido a un estímulo mecánico. Se solía analizar la picazón junto al sentido del dolor, pero ahora es considerada una nueva submodalidad ya que el camino por el que pasa la información de la piel al cerebro es independiente.

Además de su función sensorial, la piel actúa como una barrera protectora y también actúa en la regulación de la temperatura corporal. Contiene más de dos millones de glándulas sudoríparas que previenen el calentamiento excesivo del cuerpo, y alrededor de cinco millones de pelos.

Existen dos tipos principales de piel: la piel vellosa (hairy skin) que cubre la mayor parte del cuerpo, incluyendo la parte dorsal de la mano; y la piel sin vello (hairless skin) que cubre la palma de la mano y la planta de los pies.

La piel vellosa contiene vellos suaves y finos que cumplen una función absorbente para alejar la transpiración de la piel y ayudan al aislamiento térmico, y vellos protectores más largos, gruesos y visibles.

Ambos tipos de piel tienen la misma estructura general. Poseen una capa externa de células muertas llamada estrato córneo, y luego tres subcapas que contienen varios tipos de células incluyendo queratinocitos, células de Langerhans y melanocitos. En conjunto, estas cuatro capas de piel se conocen como epidermis.

Los receptores táctiles se encuentran en todo el cuerpo, tanto en la piel vellosa como en la sin vello, y están incrustados tanto en la epidermis como en la capa dérmica subyacente. No existen diferencias entre los tipos de piel en términos de los receptores que contienen o los estímulos que provocan sus reacciones.

Cuando manipulamos un objeto con nuestras manos, los distintos receptores de la piel son estimulados, en particular los receptores mecánicos<sup>4</sup>.

Por otra parte, la capacidad de percibir características espaciales finas, como los bordes de una placa, depende de la adaptación lenta de los receptores tipo I (SA I) que tienen una alta sensibilidad a los aspectos temporales y espaciales de los estímulos. Los aspectos temporales se refieren a la diferencia de tiempo que tiene que haber entre dos pulsos enviados a la piel para que éstos sean considerados como sucesivos y no simultáneos. Para el tacto es suficiente con una diferencia temporal de 5 milisegundos, la cual es ampliamente mejor que la visión (25 milisegundos), pero peor que la audición (0.01 milisegundos). Por el contrario, los receptores de adaptación rápida tipo I (FA I) responden cuando hay movimiento entre la piel y un objeto. Esto es importante para detectar el deslizamiento y mantener un agarre constante.

La detección de vibraciones táctiles se refiere a los cambios de la presión en la piel a través del tiempo. La sensibilidad de la piel a vibraciones ha sido medida en función de la frecuencia de éstas. Podemos detectar vibraciones entregadas perpendicularmente a la piel desde aproximadamente 0,5 Hz hasta 700 Hz. Somos más sensibles a las vibraciones entre 200 y 300 Hz y menos sensibles a frecuencias más bajas [19].

---

<sup>4</sup> El estímulo mecánico consiste en la aplicación de una fuerza sobre la piel. Supóngase que tocamos una mesa con un dedo. En este proceso nuestro dedo ejerce una fuerza sobre la mesa. De acuerdo con la tercera ley de Newton de la mecánica, la mesa reacciona y ejerce a su vez una fuerza sobre nuestro dedo, lo que representa un estímulo mecánico [20].

## 2.3 Trabajos relacionados

### Trabajos realizados en el ámbito de la investigación

Existen diferentes antecedentes de investigaciones sobre herramientas para facilitar el diseño de interacciones o experiencias hápticas. A continuación se presentará una revisión de las más relevantes.

#### Schneider, Maclean - 2014 - Improvising design with a Haptic Instrument [21]

Los investigadores Schneider y Maclean trabajaron en el desarrollo de una herramienta llamada mHIVE (Mobile Haptic Instrument for Vibrotactile Exploration). Esta herramienta plantea una comparativa entre la música y háptica. Su principal motivación es que una de las grandes dificultades del diseño de interfaces hápticas es la falta de herramientas que permitan un desarrollo colaborativo o con las cuales el desarrollador pueda experimentar y compartir sus avances.

A diferencia de la música, donde múltiples personas pueden estar escuchando el mismo sonido, las vibraciones o pulsaciones relacionadas con el tacto no pueden ser compartidas en simultáneo.

A su vez, con háptica tampoco tenemos un modelo conceptual unificado, es decir una convención o teoría que nos diga cómo se va a sentir la persona con la vibración diseñada, como por ejemplo lo que sucede con los acordes menores en la música y cómo esto genera una melodía que las personas sienten como triste, lo cual hace que sea fundamental la colaboración a la hora del diseño de vibraciones.

**MHIVE:** Es un instrumento háptico controlado a través de una tableta, sobre la cual el usuario del instrumento dibuja una forma que será traducida a vibración. Posee 2 actuadores que permiten a 2 personas en simultáneo sentir la misma vibración creada por la persona que controla la tableta. La interfaz permite controlar la amplitud, frecuencia, forma, ritmo, duración y estructura de la vibración.

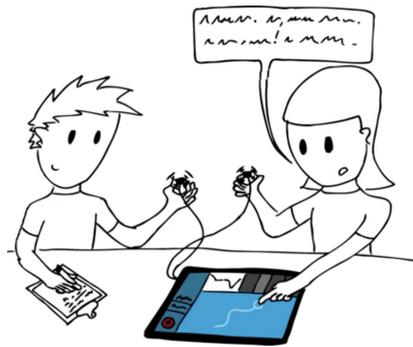
La forma dibujada se traduce con una librería de sintetización de sonido, que luego es enviada a los actuadores conectados a la entrada de audio jack de la tableta.

Los investigadores realizaron un estudio que incluía a 4 personas, para probar su experiencia con el uso de la herramienta. Todas tenían experiencia con el uso de tecnologías hápticas. Los participantes pudieron explorar la herramienta y tratar de crear sensaciones “suaves” o “ásperas”.

Como resultado del estudio, concluyeron que mHIVE permite a los usuarios o diseñadores experimentar con la creación de distintas vibraciones, ya que su interfaz es fácil de usar y eso permite que las personas puedan realizar múltiples pruebas y crear así vibraciones que no esperaban. También favorece la comunicación gracias a que varias personas pueden percibir la misma vibración, generando así un diálogo sobre cómo se sintió cada persona y permitiendo colaboración en la creación de otras vibraciones. Notaron

que aun la herramienta necesitaba de algunas funcionalidades básicas como la posibilidad de modificar y/o almacenar vibraciones, para que luego puedan ser reutilizadas o comparadas. Debido a estos faltantes, los investigadores concluyeron en que mHIVE no sirve como una herramienta de edición de ediciones hápticas.

La figura 1 ilustra el experimento realizado con el instrumento háptico, y cómo los participantes podían utilizarlo para sentir un efecto vibratorio al mismo tiempo.



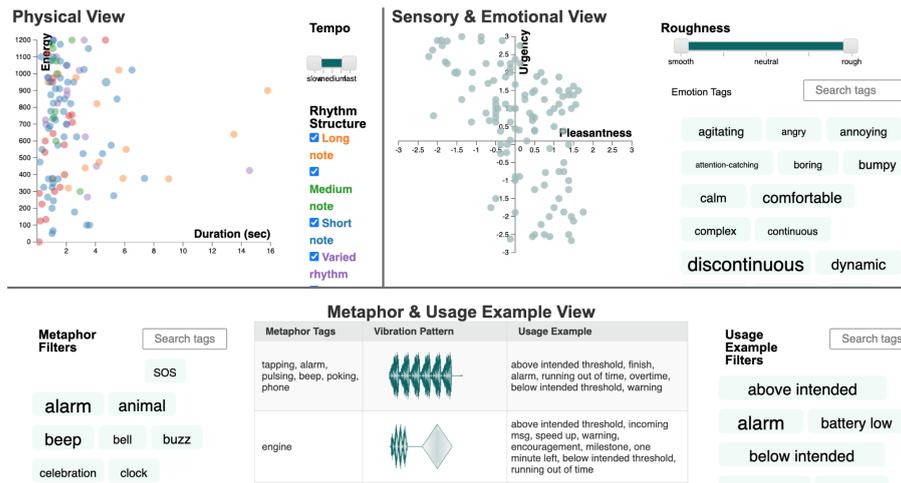
**Figura 1:** Ilustración del experimento realizado con el instrumento háptico mHIVE [22].

## Seifi, Zhang, MacLean - 2015 - VibViz Organizing, visualizing and navigating vibration libraries [23]

El trabajo desarrollado por los investigadores Hasti Seifi, Kailun Zhang and Karon E. MacLean, tiene como motivación principal la falta de uso o familiarización de los usuarios finales de las aplicaciones en cuanto a las vibraciones, provocando que muchos quieran incluso desactivar estas funciones. Plantean que las diferencias en la percepción háptica y preferencias de las personas pueden ser la causa principal de esta búsqueda de desactivación. Como posible solución plantean que es posible que el usuario final no desee desactivar la funcionalidad si tiene la posibilidad de personalizar o crear sus propias vibraciones.

Otro aspecto que intentan solucionar es la necesidad de un gran set de vibraciones definidas, que permita a cualquier persona familiarizarse y experimentar con las experiencias hápticas.

Trabajaron en el desarrollo de la herramienta **VibViz**, que cuenta con un set de 120 vibraciones. Basados en investigaciones previas sobre el lenguaje táctil que argumentan que las personas usualmente describen las vibraciones utilizando palabras relacionadas con los sentidos o emociones, agruparon sus vibraciones en 5 categorías: Físicas, emocionales, sensoriales, ejemplos de uso y metáforas. La herramienta intenta brindar a sus usuarios una forma fácil de encontrar lo que desean, proveyendo filtros y una organización visual clara, debido a la gran cantidad de distintas vibraciones que posee y para que sea fácil explorar su contenido. La figura 2 presenta la interfaz web de la herramienta VibViz desarrollada como parte de este estudio.



**Figura 2:** Interfaz web de la herramienta VibViz [24].

Los investigadores realizaron un pequeño estudio de 12 participantes (la mayoría sin experiencia en háptica), quienes utilizaron la interfaz de la herramienta y podían probar las vibraciones a través del uso de una pulsera. Como resultado del estudio notaron que su herramienta presentaba ciertos problemas en su interfaz para los usuarios novatos, y algunas funciones no fueron descubiertas naturalmente, pero más allá de eso permitía la navegación y los usuarios podían “por accidente” descubrir vibraciones. Notaron que la funcionalidad de modificación de vibraciones hubiese sido muy útil, según comentarios de los participantes.

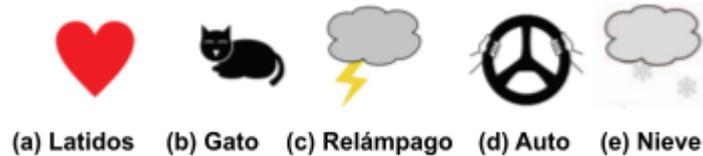
Los participantes encontraron mayor interés por las categorías “sensorial” y “emocional”, aunque en general todas resultaron interesantes. Los resultados revelaron la importancia del escaneo visual (visión general tabular y espacial, y patrón visual VT) para una navegación eficiente en la biblioteca.

Schneider, MacLean - 2016 - Studying design process and example use with Macaron, a web-based vibrotactile effect editor [25]

Oliver S. Schneider y Karon E. MacLean desarrollaron una herramienta llamada Macaron. Macaron es una herramienta web que presenta un pequeño conjunto de vibraciones definidas a modo de esqueleto, y permite al usuario o diseñador modificarlas, o bien partir desde cero para crear su propia vibración de manera interactiva, es decir que tienen la posibilidad de probar cada avance realizado. Los parámetros modificables son la amplitud y frecuencia. La vibración creada puede ser probada y reproducida a través de los parlantes del dispositivo. El usuario de la herramienta puede descargar sus vibraciones a un archivo y luego volver a importarlas para modificarlas.

La principal motivación para el desarrollo de esta herramienta fue la inexistencia de otras que permitan a los diseñadores poder crear las vibraciones y modificarlas por ellos mismos.

Realizaron un estudio con 13 participantes sin conocimientos. Se les presentó 5 animaciones y ellos debían representar lo que cada una de estas debería transmitir. Podían usar si lo deseaban los ejemplos base que propone la herramienta. En la figura 3 se ilustran estas animaciones.



**Figura 3:** Animaciones que debían recrear los participantes del estudio [25].

Como devolución de los participantes obtuvieron que es una herramienta muy fácil de utilizar, aún para gente no experta. También se les solicitaron algunas funciones extra como hacer zoom y ver el valor del parámetro (frecuencia o amplitud) sobre el gráfico y poder hacer modificaciones más específicas sobre estos. A su vez, los ejemplos provistos no fueron utilizados por los participantes ya que no coincidían con lo que los participantes debían hacer, con lo que notaron que tener un set de ejemplos como el que provee VibViz sería de gran valor.

## Demonstration-based vibrotactile pattern authoring - 2013 [26]

Kyungpyo Hong, Jaebong Lee y Seungmoon Choi desarrollaron un método para diseñar patrones vibrotáctiles basado en la interacción del usuario final, donde el dispositivo genera un patrón imitando su comportamiento.

Este método plantea una alternativa al diseño de vibraciones utilizando herramientas como Haptic Studio (Immersion Corp), las cuales requieren manipular formas (waveforms) que luego se transformarán en una vibración. Este mecanismo puede consumir mucho tiempo, ser no intuitivo y difícil para personas que no poseen muchos conocimientos.

Trabajaron en el desarrollo de una herramienta que genera una vibración diseñada por el usuario del sistema en base a como toca la pantalla. La duración de la presión se traduce a la duración de cada patrón de la vibración, la fuerza aplicada sobre la pantalla a la intensidad y la coordenada vertical a la frecuencia.

Realizaron un estudio con 10 personas sin experiencia, donde los participantes debían cumplir con 2 tareas. La primera era implementar dos “íconos táctiles” [27] y la segunda imitar una parte de una canción. Ambas tareas se realizan en dispositivos Android usando una herramienta basada en formas y otra herramienta basada en la interacción.

Como resultado del estudio pudieron notar que si bien su herramienta presenta muchas más facilidades de uso y requiere menos tiempo para crear vibraciones, puede no ser tan precisa. Los participantes encontraron que fue muy útil para imitar la canción, pero no así la primera tarea ya que requería tener un mayor control sobre el diseño.

El método de diseño de patrones vibrotáctiles basado en la interacción del usuario puede a su vez ser útil para el diseño de largos patrones, ya que permite mayor agilidad en el desarrollo y no exige la memoria del diseñador forzando a que recuerde el patrón desde el inicio.

## Toward Affective Handles for Tuning Vibrations [28]

Hastei Seifi, Mattew Chun y Karon Maclean realizaron un trabajo de investigación que buscaba demostrar una correlación entre determinados sentimientos, sensaciones o emociones que podrían llegar a generar una vibración, y qué parámetros deben configurarse para alterar estas sensaciones. En particular el estudio se basa en 3 emociones: agitación, vivacidad y rareza.

La principal motivación para esta investigación fue que muchas veces, los diseñadores crean vibraciones con el fin de producir una emoción en el usuario final (por ejemplo diseñadores de videojuegos), con lo cual para ellos podría ser mucho más sencillo o intuitivo querer que una vibración sea más “vivaz” o “alegre”, en lugar de subir su “frecuencia” o “forma” (parámetros comúnmente presentes en las herramientas de diseño de vibraciones existentes).

Como paso previo a las emociones, realizaron una abstracción de los atributos como “forma”, “frecuencia”, “amplitud” y su combinación para generar 5 parámetros que llamaron **atributos sensoriales**, y a los que luego relacionaban directamente a las emociones. Estos atributos son detallados en la Tabla 1, junto con la implementación que realizaron de éstos para realizar los estudios.

Atributo sensorial	Implementación
Energía y aspereza	Estos dos parámetros sensoriales están acoplados en la mayoría de los actuadores de vibración, y se relacionan con la agitación. La energía de una vibración está relacionada con su frecuencia y forma de onda. Para el estudio se aumentó la frecuencia y se cambió a una forma de onda cuadrada.
Tempo o Ritmo	Se utilizaron similitudes (en parámetros de diseño y hardware) reportadas para estímulos de audio y táctiles. Basándose en el algoritmo utilizado para aumentar el tempo en los archivos de sonido, redujeron la duración de los pulsos y los silencios sin afectar su tono (frecuencia).
Discontinuidad	Se examinaron cualitativamente las vibraciones discontinuas en la biblioteca VibViz, luego se modificó el número y la duración de los silencios en las vibraciones base y fueron evaluadas en estudios piloto hasta que llegaron a la siguiente implementación: Para vibraciones discontinuas, reemplazaron partes de cada pulso con silencios. Para vibraciones continuas, se dividió la vibración en secciones iguales y se reemplazó una parte de cada sección con silencio.
Irregularidad	Siguiendo un proceso similar a la discontinuidad, se encontraron vínculos entre la irregularidad y las variaciones de ritmo. Así, se añadieron silencios con una duración aleatoria a los silencios existentes en vibraciones discontinuas o a posiciones aleatorias en vibraciones continuas.
Dinamismo	Basándose en las vibraciones de VibViz y los estudios piloto, el dinamismo fue definido como una variación de amplitud y una disminución periódica de la amplitud de los pulsos.

**Tabla 1:** Presentación de atributos sensoriales, con su implementación para los estudios.

Los autores realizaron dos casos de estudios, donde seleccionaron un conjunto de vibraciones existentes en la herramienta VibViz [23] y generaron algunas vibraciones derivadas modificando estos atributos sensoriales en busca de que generen en mayor o menor medida cierta emoción. Se le presentó a los participantes la vibración base, y luego esta misma con alteraciones, para poder realizar una comparación.

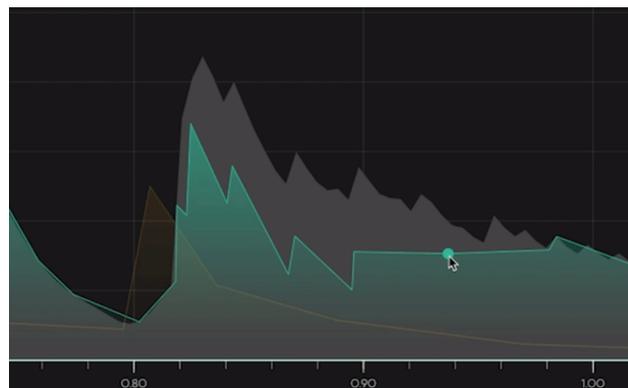
## Aplicaciones comerciales

Además de los trabajos de investigación presentados anteriormente, están comenzando a aparecer herramientas comerciales para ayudar sobre todo a ingenieros de software a incluir efectos hápticos en sus aplicaciones. A continuación se describen dos de las herramientas más utilizadas: Lofelt Studio para Android y iOS y Nice Vibrations de Mountain Studios para Unity.

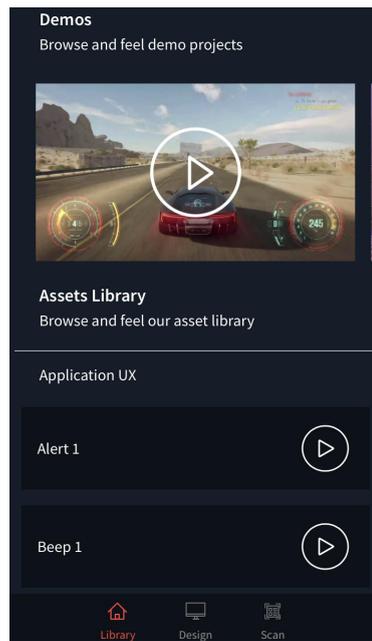
### Lofelt Studio [29]

Lofelt Studio es un conjunto de herramientas para todas las etapas del desarrollo háptico, incluida la creación, prueba e integración. Este conjunto de herramientas le permite generar automáticamente efectos hápticos realizando un análisis detallado de sus archivos de audio.

Consta de una aplicación de escritorio, una aplicación web (que provee un conjunto reducido de las funcionalidades que provee la aplicación de escritorio) y una aplicación móvil que puede ser usada para probar las vibraciones creadas en las demás aplicaciones. En la figura 4 se puede ver la herramienta web, y en la figura 5 la aplicación móvil.



**Figura 4:** Herramienta web para diseño de vibraciones de Lofelt Studio [29].



**Figura 5:** Captura de pantalla de la aplicación web para probar vibraciones de Lofelt Studio.

La aplicación de escritorio Studio se utiliza para crear clips hápticos. Estos archivos, con la extensión `.haptic`, contienen datos hápticos derivados del audio de origen proporcionado. Por ejemplo, si se carga un archivo de audio de pasos de una persona, la aplicación de escritorio generará un clip háptico acorde.

La aplicación ofrece un conjunto de vibraciones de muestra que pueden utilizarse, y también ofrece formas de refinar la experiencia háptica. Se pueden modificar los parámetros del análisis para cambiar los resultados generales, e incluso editar manualmente los hápticos a través de una interfaz gráfica que resultará familiar a los usuarios de aplicaciones de audio.

También proveen librerías, SDKs y plugins, que pueden ser útiles para integrar estas herramientas en una aplicación móvil. Al momento de la escritura de esta Tesina, el SDK de Android se encuentra en Beta.

La herramienta está pensada principalmente para juegos, donde se intenta generar una experiencia reproduciendo audio y generando efectos hápticos que acompañen ese sonido. A diferencia de la herramienta que se plantea construir en este trabajo, no está pensada para diseñadores de háptica en general, ni para poder crear efectos totalmente personalizables en respuesta a interacciones del usuario final, ni para tener efectos hápticos sin reproducción de audio.

Otra diferencia clara en relación a la herramienta que se busca construir con este trabajo, es que los diseñadores deberán aprender a utilizar otra herramienta nueva, extra a las que ya conocen y utilizan para diseñar interfaces, y este trabajo busca que puedan incluir sus vibraciones en su flujo de trabajo actual.

## Nice Vibrations [30]

Nice Vibrations es una solución simple pero poderosa para agregar vibraciones y retroalimentación háptica en juegos. Ofrece una interfaz universal para joysticks, iOS y Android a la vez. La herramienta cuenta con varias vibraciones preestablecidas que pueden ser utilizadas, y además permite alterar ciertos parámetros para generar una vibración personalizada. Al utilizar las APIs de vibración de los dispositivos, posee algunas limitaciones relacionadas a éstas, como por ejemplo la imposibilidad de controlar la magnitud de la vibración para dispositivos Android.

Esta herramienta puede ser utilizada únicamente para desarrollos con Unity [31], que es una plataforma para desarrollo de videojuegos. Además, a diferencia de la herramienta que este trabajo propone, es necesario que un programador genere un prototipo de la vibración para poder probarla en el dispositivo que se utilizará.

## 2.4 Diseño de experiencia de usuario<sup>5</sup>

### Definiciones básicas y proceso

En esta sección introduciremos las definiciones básicas asociadas al diseño de experiencias de usuario y los distintos aspectos que éste abarca. También hablaremos sobre los procesos que siguen los diseñadores, y la importancia de la participación de los usuarios en ellos.

### Diseño de interacciones

Por diseño de interacción, nos referimos a diseñar productos interactivos para respaldar la forma en que las personas se comunican e interactúan en su vida diaria y laboral. Se trata de crear experiencias de usuario que mejoren y aumenten la forma en que las personas trabajan, se comunican e interactúan.

El término diseño de interacción se utiliza generalmente para describir el campo, incluidos sus métodos, teorías y enfoques. UX se usa más ampliamente en la industria para referirse a la profesión. Sin embargo, los términos se pueden usar indistintamente. Además, depende de su ética y marca.

La experiencia de usuario trata de cómo un producto se comporta y cómo es usado por los usuarios. Jakob Nielsen y Donan Norman (gurús de UX) consideran que la experiencia de usuario engloba todos los aspectos de la interacción de los usuarios finales con una compañía, sus servicios y productos. Se trata de la satisfacción de las personas al usar un producto, mirarlo, sostenerlo, abrirlo o cerrarlo. Incluye la impresión general de lo bueno que es usarlo, hasta el efecto que tienen los pequeños detalles, como la suavidad con la que gira un interruptor, el sonido de un clic o el toque de un botón al presionarlo.

---

<sup>5</sup> Esta sección corresponde a un resumen de los conceptos más relevantes del libro Interaction Design - Beyond Human-Computer Interaction (2019) [32]

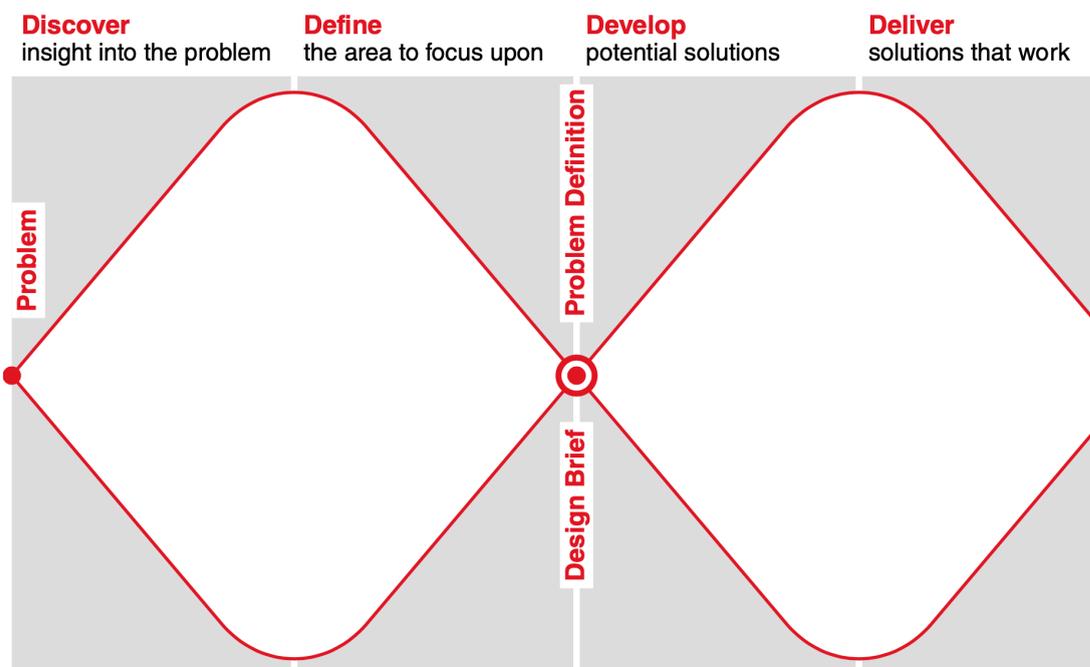
A veces los diseñadores hacen referencia a UX (user experience) como UXD, alentando el uso de un paradigma de design thinking que se enfoque en la calidad de la experiencia, por sobre el conjunto de métodos usados.

Hay muchos aspectos de la experiencia del usuario a considerar y tener en cuenta al diseñar productos interactivos. De importancia central son la usabilidad, la funcionalidad, la estética, el contenido, la apariencia y el atractivo emocional. Además, Jack Carroll destaca otros aspectos, como la diversión, la salud, el capital social (los recursos sociales que se desarrollan y se mantienen a través de las redes sociales, valores compartidos, metas y normas) e identidad cultural, como edad, etnia, raza, discapacidad, estado familiar, ocupación y educación.

El modelo de la experiencia de usuario que plantea Marc Hassenzahl es el más conocido, donde lo conceptualiza en términos de aspectos pragmáticos y hedónicos. Por pragmático, se entiende cuán simple, práctico y obvio es para el usuario lograr sus objetivos. Por hedónico, se entiende cuán evocadora y estimulante es la interacción para ellos.

## Aspectos que abarca el diseño de interacciones

Hay muchos campos del diseño, como el diseño gráfico, el diseño arquitectónico, el diseño industrial y el diseño de software. Aunque cada disciplina tiene su propio enfoque, hay puntos en común. El Design Council del Reino Unido captura estos puntos en lo que llaman el doble diamante del diseño [33]. Este enfoque tiene cuatro fases que se repiten, ilustradas en la Figura 6.



**Figura 6:** Doble diamante del diseño, corresponde a ilustración del libro Interaction Design - Beyond Human-Computer Interaction (2019) [32].

- Descubrir: Los diseñadores intentan recopilar información sobre el problema.
- Definir: Los diseñadores desarrollan un resumen claro que enmarca el desafío del diseño.
- Desarrollar: Se crean prototipos, se prueban y se itera sobre soluciones o conceptos.
- Entregar: El proyecto resultante se finaliza, se produce y se lanza.

El diseño de interacción también sigue este proceso y está respaldado por la filosofía del diseño centrado en el usuario, es decir, involucrar a los usuarios durante todo el desarrollo. Tradicionalmente, los diseñadores de interacción comienzan haciendo una investigación del usuario y luego generando bocetos de sus ideas.

## Descubriendo el problema

En el proceso de creación de un producto interactivo, puede ser tentador querer descubrir cómo diseñar la interfaz física y qué tecnologías y estilos de interacción usar, por ejemplo, si usar una pantalla multitáctil, voz, interfaz gráfica de usuario, pantalla de visualización frontal, realidad aumentada, basada en gestos y así sucesivamente. El problema de comenzar de esa forma es que los potenciales usuarios y su contexto pueden ser malinterpretados, y los objetivos de usabilidad y experiencia del usuario pueden pasarse por alto. Si bien es cierto que en algún momento es necesario elegir qué tecnología emplear y decidir cómo diseñar los aspectos físicos, es mejor tomar estas decisiones después de comprender la naturaleza del problema, es decir, comprender cuál es actualmente la experiencia del usuario o el producto, por qué se necesita un cambio y cómo este cambio mejorará la experiencia del usuario. Incluso cuando se diseña para una nueva experiencia de usuario, también es necesario comprender el contexto para el que se utilizará y las posibles expectativas de los usuarios actuales.

El proceso de descubrimiento del problema se realiza como un esfuerzo de equipo. Los miembros del equipo suelen tener diferentes perspectivas al respecto. Por ejemplo, es probable que un gerente de proyecto esté preocupado por una solución propuesta en términos de presupuestos, plazos y costos de personal, mientras que un ingeniero de software estará pensando en dividirla en conceptos técnicos específicos. Las implicaciones de perseguir cada perspectiva deben considerarse en relación con las demás. Aunque lleva mucho tiempo y a veces da lugar a desacuerdos entre el equipo de diseño, los beneficios de este proceso pueden superar con creces los costos asociados: habrá muchas menos posibilidades de que las suposiciones incorrectas y las afirmaciones no respaldadas se introduzcan en una solución de diseño que luego resulte inutilizable o no deseada.

## La importancia de involucrar a los usuarios

Involucrar a los usuarios en el desarrollo es importante porque es la mejor manera de garantizar que el producto final sea utilizable y que, de hecho, se utilizará. Esta

estrategia permite asegurar que los desarrolladores comprendan bien los objetivos de los usuarios.

Existen otros dos aspectos a tener en cuenta para asegurar la usabilidad del producto:

## Manejo de expectativas

Es el proceso de asegurarse que las expectativas del usuario sobre el producto son realistas, y evitar que el usuario se sorprenda o se sienta engañado al recibir el producto. Involucrar al usuario en el proceso de desarrollo puede ayudar a prevenir esto ya que podrá ver los avances del producto en etapas tempranas o intermedias de su desarrollo.

Si los usuarios tienen la oportunidad de trabajar con el producto antes de su lanzamiento, mediante capacitación o demostraciones prácticas de una versión preliminar, comprenderán mejor qué esperar cuando el producto final esté disponible.

## Sentido de propiedad

Los usuarios que están involucrados y sienten que han contribuido al desarrollo de un producto tienen más probabilidades de tener un sentido de propiedad hacia él y apoyar su uso.

La participación de los usuarios puede darse a distintas escalas, desde la inclusión en todas las iteraciones del proceso de desarrollo o solo en algunas actividades, o también pueden incluirse tanto a grandes grupos de usuarios como pequeños grupos o solo una persona.

Los usuarios individuales pueden ser incorporados al equipo de diseño para que sean los principales contribuyentes al desarrollo. Una consecuencia negativa podría ser que la participación a tiempo completo puede significar la pérdida de contacto con su comunidad de usuarios, mientras que la participación a tiempo parcial puede resultar en una gran carga de trabajo para ellos. En el lado positivo, tener un usuario involucrado a tiempo completo o parcial significa que el intercambio de ideas, opiniones y devoluciones es continuo durante todo el desarrollo. La participación de los usuarios en actividades específicas para informar el desarrollo o evaluar diseños una vez que estén disponibles puede también ser una forma valiosa de participación, pero el aporte de los usuarios se limita a esa actividad en particular.

## Diseño centrado en el usuario

Cuando fue establecido el campo de la interacción humana con la computadora (HCI), John Gould y Clayton Lewis establecieron tres principios que creían pueden conducir a un "sistema informático útil y fácil de usar". Estos principios son aceptados como las bases de un diseño centrado en el usuario.

## Enfoque temprano en usuarios y tareas

Requiere comprender quiénes serán los usuarios mediante el estudio directo de sus características cognitivas, conductuales, antropomórficas y actitudinales. Es necesario observar a los usuarios haciendo sus tareas normales, estudiar la naturaleza de esas tareas y luego involucrar a los usuarios en el proceso de diseño.

Cuando la gente tiene problemas con la tecnología, suele pensar que es culpa suya. Las personas son propensas a cometer errores y tienen ciertas limitaciones, tanto cognitivas como físicas. Los productos diseñados para ayudar a las personas deben tener en cuenta estas limitaciones y tratar de evitar que se cometan errores

Todas las decisiones de diseño se toman dentro del contexto de los usuarios, sus actividades y su entorno.

## Medición empírica

Al principio del desarrollo, se observan y miden las reacciones y el desempeño de los usuarios ante escenarios impresos, manuales, etc. Posteriormente, los usuarios interactúan con simulaciones y prototipos, y su desempeño y reacciones se observan, registran y analizan nuevamente.

Siempre que sea posible, los objetivos específicos de usabilidad y experiencia del usuario deben identificarse, documentarse claramente y acordarse al comienzo del proyecto. Esto puede ayudar a los diseñadores a elegir entre diseños alternativos y verificar el progreso a medida que se desarrolla el producto. La identificación de objetivos específicos por adelantado significa que el producto se puede evaluar empíricamente en etapas regulares durante el desarrollo.

## Diseño interactivo

Cuando se encuentran problemas en las pruebas de usuario, se solucionan y luego se realizan más pruebas y observaciones para ver los efectos de las correcciones. Esto significa que el diseño y el desarrollo son iterativos, con ciclos de diseño-prueba-medida-rediseño que se repiten tantas veces como sea necesario.

## Actividades básicas del diseño de interacción

A continuación presentaremos las actividades básicas que siguen los diseñadores a la hora de trabajar en un diseño de interacción y cómo éstas se relacionan con el doble diamante del diseño presentado anteriormente, destacando también la importancia de la generación de prototipos en este proceso.

## Descubrimiento de requerimientos

Esta actividad cubre el lado izquierdo del doble diamante del diseño. Se centra en descubrir algo nuevo sobre el mundo y definir lo que se desarrollará. En el caso del diseño de interacción, esto incluye la comprensión de los usuarios y el apoyo que un producto interactivo podría proporcionar de manera útil. Esta comprensión se obtiene a través de la recopilación y el análisis de datos. Constituye la base de los requisitos del producto y sustenta el diseño y desarrollo posteriores.

## Diseñando alternativas

Es la actividad principal del diseño y es parte de la fase de desarrollo del doble diamante. Consiste en proponer ideas que cumplan con los requerimientos y puede verse en dos partes, diseño conceptual y diseño concreto. El diseño conceptual incluye la construcción de un modelo conceptual del producto, de lo que las personas podrían hacer con él, y aquellos conceptos necesarios para interactuar con él. El diseño concreto considera los detalles del producto, incluyendo colores, sonidos, imágenes, íconos, etc.

## Prototipado

Es parte de la fase de desarrollo del doble diamante. El diseño interactivo involucra también el diseño del comportamiento de productos interactivos, y la forma más eficiente para que los usuarios puedan evaluar estos diseños, es mediante interacciones con ellos. Esto no necesariamente requiere desarrollo de software.

Un prototipo es una manifestación de un diseño que permite a los stakeholders interactuar con él y explorarlo. Generalmente se utiliza para hacer énfasis en algunas características del producto, por sobre otras. Adoptan muchas formas, por ejemplo, un modelo a escala de un edificio, un puente o una pieza de software que se rompe al cabo de pocos minutos.

Los prototipos pueden ser muy útiles para discutir o evaluar ideas con los stakeholders, comunicar algo a los miembros del equipo o incluso pueden ser utilizados por los diseñadores para explorar nuevas ideas. Sirven para responder preguntas y dar soporte a los diseñadores en la elección de alternativas. Su objetivo principal suele ser permitir pruebas rápidas de un aspecto particular del producto.

Dan Saffer distingue entre un prototipo de producto y un prototipo de servicio, donde este último involucra juegos de rol y personas como parte integral del prototipo, así como del producto en sí.

Es común que los prototipos evolucionen a través de varias etapas de fidelidad, dentro de los ciclos de diseño-evaluación-rediseño. Inicialmente se suele contar con prototipos de baja fidelidad, que no se parecen mucho al producto final e incluso podrían no tener funcionalidades asociadas. Son útiles ya que suelen ser simples, baratos y fáciles de producir y modificar, lo que permite la exploración de nuevas ideas, aunque no suelen

formar parte del producto final. También los prototipos pueden ser de alta fidelidad, que suelen parecerse mucho más al producto final y pueden proveer varias funcionalidades. Se pueden desarrollar prototipos de alta fidelidad modificando e integrando componentes existentes, tanto hardware como software.

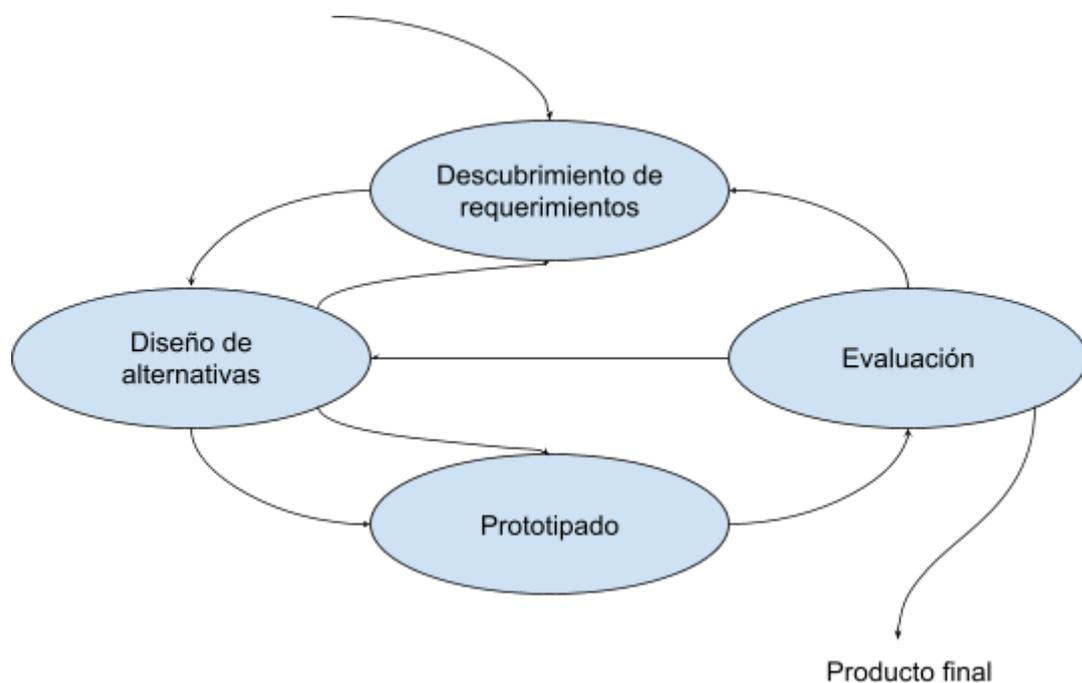
Dos propiedades de los prototipos que también nos permiten caracterizarlos son la amplitud de la funcionalidad frente a la profundidad. Estos dos tipos de prototipos se denominan prototipos horizontales (que proporcionan una amplia gama de funciones pero con pocos detalles) y prototipos verticales (que proporcionan muchos detalles para unas pocas funciones). Otra característica es el nivel de robustez frente al grado de variabilidad. Hacer un prototipo robusto puede hacer que sea más difícil de cambiar.

El tipo de preguntas que puede responder cualquier prototipo es limitado, y el prototipo debe construirse teniendo en cuenta las cuestiones que se desean destacar. En la creación de prototipos de baja fidelidad, está bastante claro que se han hecho concesiones.

## Evaluación

Es parte de la fase de desarrollo del doble diamante. Es el proceso de determinar la usabilidad y aceptabilidad del producto o diseño, medido en términos de una variedad de criterios de usabilidad y experiencia del usuario.

Todas estas actividades se encuentran interrelacionadas entre sí, y constituyen partes importantes en el ciclo de diseño de interacciones. Esto se encuentra ilustrado en la figura 7.



**Figura 7:** Actividades del diseño de interacciones y su interacción para la construcción de un producto final

## Herramientas de prototipado [34]

La parte central de todo el proceso de diseño de Experiencias de Usuario es el prototipado. La construcción de representaciones sobre el flujo de una aplicación, las transiciones de una pantalla a otra, algo que no solo resulta útil para un equipo de desarrolladores y diseñadores de aplicaciones, sino también cuando se busca recaudar fondos o validar los requerimientos de un cliente antes de efectivamente iniciar el desarrollo.

Existe una gran cantidad de aplicaciones de prototipado de interfaces, tanto para aplicaciones web como móviles, o ambas. A continuación describiremos algunas de ellas y sus principales funciones.

## Sketch - the digital design toolkit [35]

La herramienta de diseño de vectores específicos facilita que los diseñadores no piensen en las resoluciones o las densidades de pantalla cuando crean maquetas, logotipos y diseños. La herramienta viene con capacidades de edición de imágenes muy poderosas que la hacen tan capaz como Photoshop.

Entre sus mejores funciones se destaca el conjunto de símbolos que viene con la herramienta, que permite prediseñar elementos como botones o widgets, y también la funcionalidad de “exportar a código”.

En la Figura 8 se puede ver una captura de la interfaz de Sketch, junto con un ejemplo diseñado en dicha herramienta.

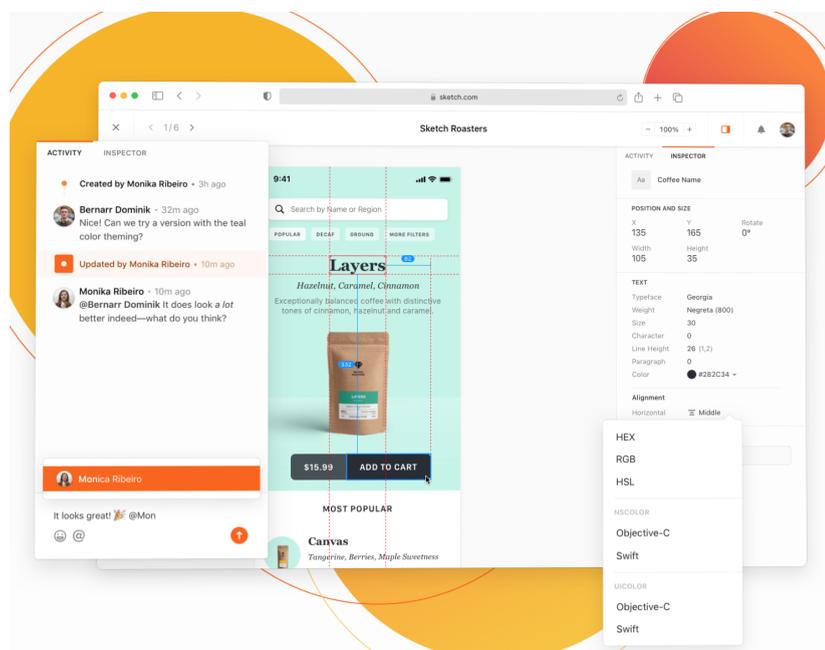


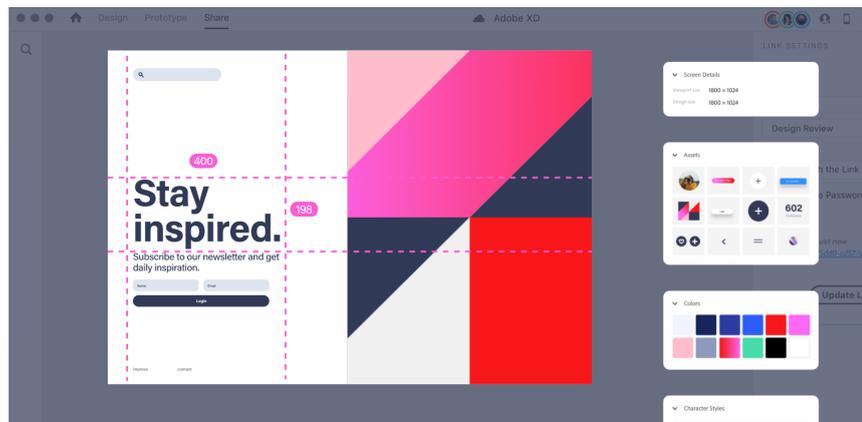
Figura 8: Captura de Sketch.

## Adobe Experience Design (Adobe XD) [36]

La herramienta viene con todas las funcionalidades que están presentes en Sketch y tiene una interfaz muy intuitiva. Adobe XD incluye además una funcionalidad que permite arrastrar y soltar elementos dentro de la herramienta, los cuales luego pueden ser agregados a los diseños.

Esta herramienta está integrada con el resto de los productos que posee Adobe, con lo cual los diseñadores pueden adaptar en esta herramienta sus diseños o elementos provenientes de Photoshop o Illustrator.

La Figura 9 ilustra la interfaz de diseño de la herramienta Adobe XD.



**Figura 9:** Captura de Adobe XD.

## Balsamiq [37]

Uno de los mayores argumentos de venta de Balsamiq es la velocidad de creación de prototipos que permite al diseñador de UX esbozar conceptos rápidamente. Balsamiq permite arrastrar y luego posicionar los diferentes elementos, al mismo tiempo que brinda a los diseñadores la misma experiencia que obtendrían al trabajar en pizarras blancas.

En la Figura 10 se muestra la herramienta, junto con un diseño realizado en ella.

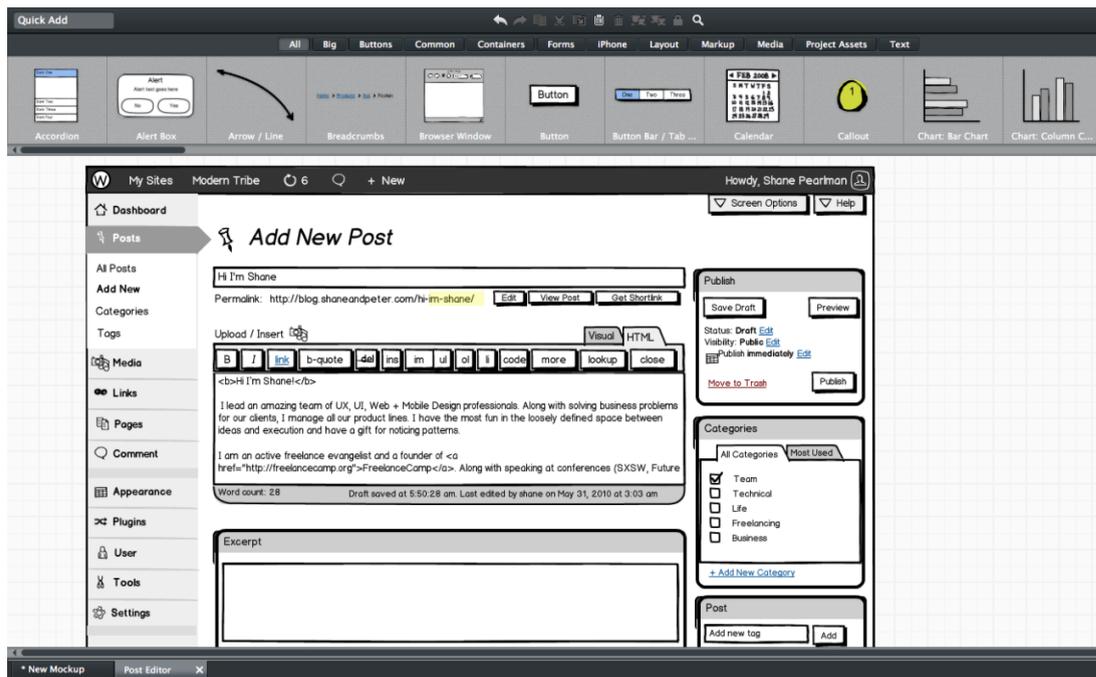


Figura 10: Ejemplo de diseño en Balsamiq [38].

## Figma [39]

Es una de las herramientas de diseño más utilizadas actualmente. Se destaca por sus funciones de trabajo colaborativo y facilidad de uso. Además posee interfaces muy intuitivas y la posibilidad de ordenar el trabajo en componentes que pueden ser reutilizados a lo largo de todo el diseño y distintas pantallas. La Figura 11 muestra la interfaz de Figma.

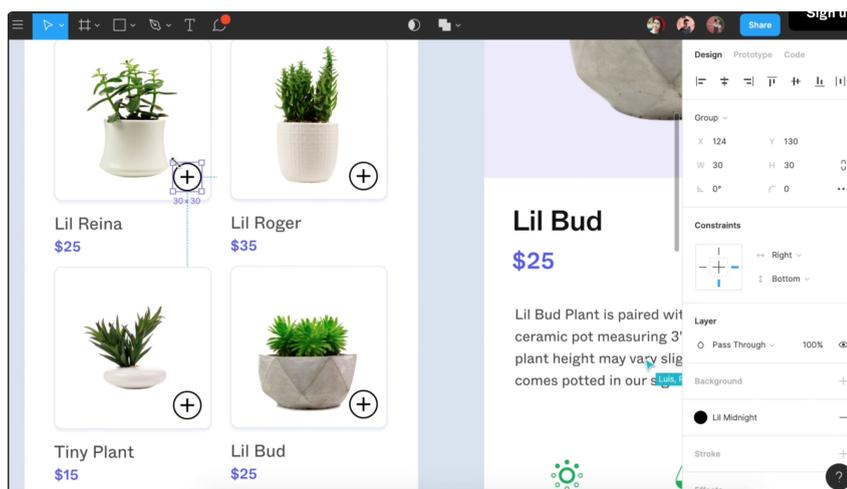


Figura 11: Captura de diseño con Figma.

## Framer [40]

La herramienta permite desarrollar elaborados flujos de experiencias de usuario con textos y formas. Simplifica el desarrollo responsive para todos los tamaños, manteniendo los elementos intactos. Para funcionalidades complejas permite diseñar componentes utilizando código y de esa forma generar interfaces más dinámicas.

En la Figura 12, se puede ver la interfaz de Framer y un diseño en la herramienta.

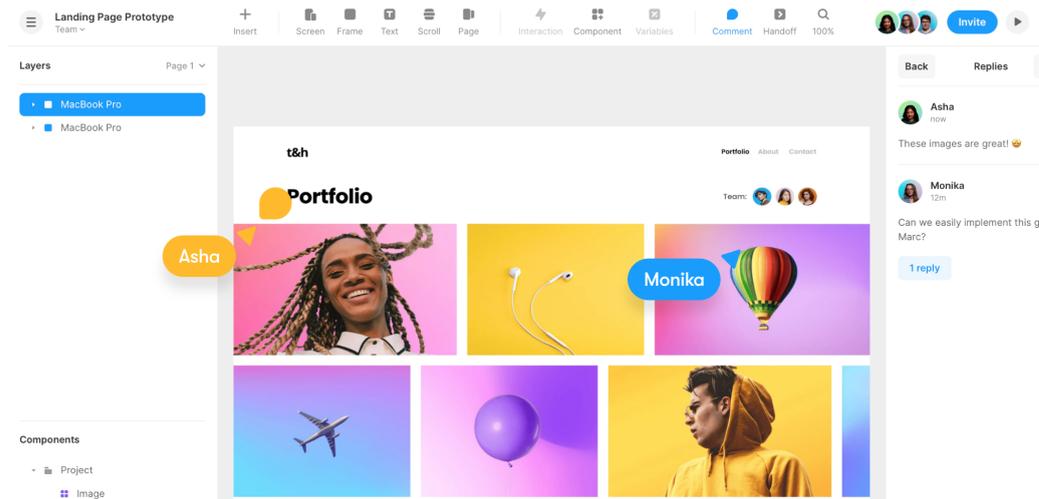


Figura 12: Captura de diseño con Framer.

En el capítulo [Propuesta](#) se hará un análisis más en detalle de cada una de estas herramientas y algunas pruebas de concepto para mostrar la utilidad de éstas en la herramienta que se busca desarrollar con este trabajo.

## Diseño de experiencias hápticas

Después de presentar los conceptos fundamentales de háptica y de diseño de UX y relevar algunas de las herramientas disponibles para ambas tareas (tanto en el marco académico como industrial), vamos a presentar algunos de los criterios y tareas relevantes y necesarias a la hora de hacer la integración de háptica en el diseño UX.

## Diseñando para las diferencias

Debido a que cada persona puede percibir diferente una vibración, no es algo tan sencillo diseñar una experiencia que pueda generar un efecto similar en todos los usuarios.

Existen diversas interpretaciones compartidas que fueron descubiertas por algunos investigadores. Por ejemplo, muchos individuos coinciden que la urgencia puede ser representada por vibraciones con altos valores de frecuencia y energía [28].

Es importante para los diseñadores poder partir de una vibración preexistente y poder modificarla, y de esta forma poder adaptarla para un contexto en particular.

## Proceso de diseño

En el proceso de diseño UX, generalmente basado en estrategias como design thinking, existen algunas actividades que contribuyen fuertemente a la creación y refinamiento de prototipos: como la búsqueda o browsing de ejemplos y las acciones para poner esos ejemplos a disposición de la comunidad [41].

## Búsqueda

Es normal que los diseñadores tengan un repertorio de ejemplos resultantes de trabajos anteriores, o que generaron para practicar o aprender. El diseño generalmente comienza revisando estos ejemplos o buscando en diversos sitios para conseguir inspiración.

Es importante para los diseñadores una clasificación de estos ejemplos, ya que lo que tratan de generar son experiencias que pueden ser parte de un contexto con otros elementos audiovisuales, o imitar una sensación o sentimiento. Debido a que no hay un léxico establecido, es difícil establecer estos criterios de organización de ejemplos.

## Prototipado

Como hemos visto en secciones anteriores, hacer bocetos y prototipos permite a las personas formar vistas parciales abstractas de un problema o diseño. El boceto de interfaces puede hacerse con papel y lápiz o con ayuda de software, pero dibujar el movimiento y en particular, mostrar lo que se siente o podría sentirse una experiencia vibrotáctil, es más complicado.

## Refinado o personalización

El diseño requiere de una iteración sobre un set inicial de ideas para generar un concepto bien definido, hasta que solo queden pequeños ajustes por realizar. La incorporación de tecnología háptica a un diseño depende del hardware, la aplicación, el software y el contexto multimodal. Este proceso puede llegar a ser muy complejo, y requiere una interacción constante del desarrollador o diseñador que necesita hacer pruebas, con lo que es de gran importancia que sea rápido y fácil de probar.

## Compartir

Compartir diseños es valioso en diferentes etapas del proceso, ya sea para recibir comentarios informales de amigos y colegas, o distribuirlos para su uso y a la comunidad para su reutilización.

En los casos que es necesario un dispositivo de hardware particular para ejecutar las vibraciones, esto puede resultar engorroso ya que este debe viajar todo el tiempo para que el diseñador pueda compartir su trabajo.

## Personalizando háptica

Los efectos vibrotáctiles fueron diseñados para comunicar información y sentimientos, sin embargo, la incorporación de esto último en háptica no es sencilla. Las respuestas afectivas a los estímulos hápticos sintéticos aún no están bien catalogadas y las diferencias individuales tanto en la percepción como el afecto, hacen que sea aún más difícil [42].

Para asegurar la efectividad de los efectos, los diseñadores utilizan el diseño interactivo y la evaluación del usuario, pero esto resulta poco efectivo cuando se trata de sentimientos.

La modificación o personalización de vibraciones preexistentes resulta ser una aproximación más práctica para usuarios sin conocimientos de diseño.

## Variabilidad de herramientas de diseño

Schneider propone 5 parámetros a lo largo de los cuales las herramientas de diseño y personalización pueden variar. Estos parámetros pueden influir en la percepción de flexibilidad y esfuerzo de los usuarios para diseñar efectos hápticos y, en consecuencia, en sus preferencias y elección de herramientas [42].

## Tamaño del espacio de diseño

El tamaño del espacio de diseño se refiere al número de estímulos distintos que la herramienta permite crear. Depende tanto de la herramienta como del dispositivo utilizado para reproducir las vibraciones.

## Granularidad del control

La unidad más pequeña de un estímulo que un usuario puede manipular directamente con una herramienta, que puede variar desde un control holístico (grueso) hasta local (fino).

## Marco de diseño proporcionado

Inevitablemente, cada herramienta de diseño impone una estructura de diseño. Esta estructura es impuesta al usuario, el cual debe familiarizarse y acostumbrarse a ese manejo del espacio de diseño.

## Parámetros facilitados

El nivel de facilidad y control que una herramienta de diseño provee sobre los parámetros que permite modificar.

## Claridad de las alternativas de diseño

Las herramientas varían en la medida en que se proporcionan diseños alternativos a los usuarios, en comparación con los que se descubren.

## Elección de herramientas

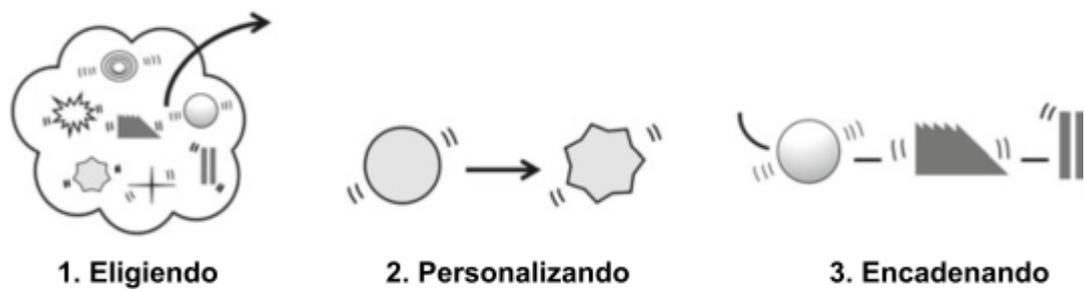
Los parámetros previamente mencionados, junto con la experiencia del diseñador en cuestión, son factores muy importantes a la hora de elegir una herramienta con la cual trabajar.

Con la presentación de éstos parámetros, Schneider plantea la siguiente relación entre ellos:

- El tamaño del espacio de diseño y la flexibilidad del marco de diseño afectan la percepción del control del diseño.
- El control holístico sobre los estímulos y la visibilidad de la alternativa de diseño puede reducir la percepción de esfuerzo.
- La preferencia es una función del control de diseño percibido, el esfuerzo y la diversión de la interfaz.

## Modelos conceptuales

Basándose en herramientas existentes, y utilizando variaciones de los parámetros propuestos, Schneider junto a Seifi crearon 3 modelos conceptuales [42]. Estos modelos son ilustrados en la Figura 13.



**Figura 13:** Modelos conceptuales de Schneider y Seifi.

1. Eligiendo: Los usuarios pueden seleccionar una vibración de una lista de vibraciones pre-existentes.
2. Personalizando: Los usuarios pueden modificar características de alto nivel de una vibración, cambiando valores.
3. Encadenando: Los usuarios cuentan con un conjunto de bloques de vibraciones, los que pueden combinar, por ejemplo secuencialmente, para crear una nueva vibración.

Los resultados de un estudio del Mago de Oz (WoZ) con prototipos en papel de las herramientas sugirieron que **Personalizando** es el enfoque preferido por ser "rápido", "eficaz" y proporcionar una sensación de "control". **Encadenando** era "divertido" pero requería "tiempo" y "buen humor", por lo que era menos práctico para escenarios cotidianos. Finalmente, **Eligiendo** fue el menos preferido por su "control" limitado, pero fue calificado como el más fácil de usar.

## 2.5 Conclusión

Luego de investigar los trabajos previos, podemos concluir que aunque han habido avances en el área, aún hay mucho camino por recorrer.

En particular, las herramientas disponibles aún ofrecen fundamentalmente el punto de vista de ingenieros de señales o de software y plantean un umbral alto de conocimientos previos para utilizarlas con solvencia. Es necesario un enfoque de inclusión de herramientas en los términos y criterios propios de los diseñadores de UX.

Por otro lado, la falta de estandarización o categorización de los efectos supone un problema que dificulta la búsqueda de efectos que satisfagan una necesidad puntual. Los avances en cuanto a categorización de vibraciones han sido notorios y pueden ser útiles al momento de buscar una vibración que se parezca a algún efecto de la vida cotidiana, como los latidos del corazón.

# Capítulo 3: Propuesta

## 3.1 Introducción

Una de las principales motivaciones de este trabajo es la falta de herramientas dedicadas al diseño de háptica para dispositivos móviles. La ventaja de trabajar con estos dispositivos es que podemos utilizar su motor de vibración, con lo cual no necesitaremos de un dispositivo de hardware adicional para poder probar nuestras vibraciones diseñadas.

El objetivo de este trabajo es entonces proveer una herramienta que facilite a los diseñadores de UX en aplicaciones móviles la inclusión de efectos hápticos vibrotáctiles en sus propuestas. Para ello creemos necesario que cualquier herramienta debe operar en los mismos criterios y parámetros que las que ya utiliza el diseñador UX y además debe incluir una amplia cantidad de muestras de distintos tipos de vibraciones, y brindar la flexibilidad para poder modificarlas y/o personalizarlas, para que de esta manera el diseño de los componentes sea más simple para el diseñador. También debe poder acoplarse con cualquier tipo de componente visual, permitiendo agregarle comportamiento vibratorio a cada botón, input<sup>6</sup>, o elemento que el usuario de la herramienta desee.

Dado que las vibraciones generadas tienen el objetivo de dar feedback al usuario como consecuencia de una acción sobre la interfaz visual, y también que los diseñadores de háptica en su mayoría están familiarizados con el uso de herramientas de diseño de interfaces, se realizó un trabajo de investigación e implementación sobre posibles herramientas que nos permitan agregar el efecto diseñado a un elemento particular del diseño de interfaces y poder interactuar con él a través de una aplicación móvil de la misma herramienta. Así el diseñador podrá diseñar por ejemplo un botón, y agregarle una vibración para cuando el usuario de la aplicación haga clic sobre éste, dentro del mismo prototipo de diseño.

El corazón de la propuesta de esta tesina es la generación de una herramienta web integrada en una de las herramientas de prototipado UX existentes, donde los diseñadores puedan crear sus vibraciones y probar el efecto generado en su dispositivo móvil en un proceso iterativo. Como la herramienta a desarrollar no se implementa como aplicación nativa para móviles, utilizaremos para generar los patrones vibratorios la API diseñada por el Consorcio WWW (W3C) implementada en los navegadores web [43], la cual presenta varias limitaciones que serán desarrolladas en profundidad en este capítulo.

## 3.2 Escenario de uso

Consideramos dos escenarios posibles con los que los diseñadores de UX que generan prototipos de UI pueden encontrarse al momento de incluir un efecto háptico para aumentar las interacciones con sus componentes visuales. Sobre la base de estos escenarios se desprenden algunos requerimientos que la herramienta a desarrollar deberá satisfacer.

---

<sup>6</sup> Elemento que requiere que el usuario ingrese información. Podría ser texto, número o un campo de selección múltiple.

## Escenario 1 - Diseñar un efecto háptico

En este escenario el diseñador tiene que crear un efecto háptico nuevo. La herramienta a proponer debería permitirle tanto generar un efecto completamente nuevo (diseñando el patrón de vibraciones completamente desde cero), como diseñar un nuevo efecto mediante la edición de uno existente. La herramienta debería facilitarle la tarea en ambos casos, mediante una visualización interactiva de las curvas de vibración a obtener.

En el caso de la edición de efectos disponibles, podría ser muy útil que la herramienta cuente con un set de vibración de ejemplo y así ayudar a los usuarios a iniciar su diseño con un umbral de acceso más bajo.

Otro requerimiento importante, además de una visualización interactiva de los patrones de frecuencia es la posibilidad de probarlo en un dispositivo móvil, que es el que dispone del hardware específico de vibración, con lo cual la herramienta debe proveer una forma de que el efecto sea llevado fácilmente a un celular.

Posterior a las pruebas, es posible que el diseñador quiera nuevamente retocar su diseño, y luego realizar pruebas otra vez. Esta iteración debe ser posible y no tener restricciones.

## Escenario 2 - Utilizar un efecto háptico de una colección

Es el caso en que el diseñador quiera utilizar un efecto creado previamente “as is” o que lo utilice como base para una nueva modificación.

La herramienta debe permitir iniciar el aumento de widgets con diseños hápticos previos obtenidos de una colección previa. O, en una combinación con el escenario 1, modificar los parámetros para ajustarlos a la situación actual.

El diseñador va a poder modificar los parámetros, afectando también al gráfico, y luego debe ser capaz de probar la vibración generada en un dispositivo móvil e iterar de la misma forma que se describe en el escenario anterior.

### 3.3 Selección de plataforma de base

Una de las principales funcionalidades que este trabajo intenta brindar es la posibilidad de que los diseñadores no solo diseñen su propia vibración, sino que también ellos mismos durante su proceso de creación puedan reproducirla en un dispositivo móvil. Con esta funcionalidad se espera reducir los tiempos de desarrollo, facilitando la creación de prototipos y su respectiva validación, sin necesidad de los desarrolladores ni de generar prototipos descartables, como también reducir el uso de nuevas herramientas, incluyendo el diseño de vibraciones como parte del flujo de trabajo normal de un diseñador de experiencias de usuario.

El efecto vibrotáctil debe producirse como consecuencia de una acción del usuario del sistema sobre una interfaz, con lo cual se utilizará una herramienta de diseño de

interfaces y se extenderá su comportamiento para poder reproducir la vibración diseñada en nuestra herramienta externa.

Se realizó una investigación teniendo en cuenta las herramientas de diseño de interfaces de usuario más utilizadas, en busca de alguna que nos permita reproducir la vibración de la manera esperada. Una de las principales condiciones que tenía que tener la herramienta es contar con una aplicación móvil para visualizar los diseños en los dispositivos móviles, ya que eso nos permitiría enfocar el trabajo en el diseño de las vibraciones, más que en el desarrollo de una aplicación nativa o herramienta externa que reproduzca la vibración, teniendo en cuenta que las computadoras no cuentan con un motor de vibración.

Asimismo la herramienta debía proveernos una forma de interactuar con el motor de vibración del dispositivo móvil. Básicamente la investigación consistía en encontrar una herramienta que cumpla ambas condiciones y que sea utilizada por la comunidad de desarrolladores, evitando que tengan que aprender a utilizar una nueva herramienta de diseño. Algunas herramientas fueron investigadas pero descartadas de inmediato por ejemplo por la ausencia de una aplicación móvil como Balsamiq [37].

## Relevamiento y estudio de Herramientas de maquetado comerciales

En esta sección se explicarán en detalle las pruebas realizadas sobre un conjunto de herramientas de prototipado, con el objetivo de utilizar dicha plataforma como herramienta donde los diseñadores puedan no solo crear una interfaz visual, sino también agregar sus vibraciones como parte del diseño y realizar pruebas interactivas en él.

### Figma

Figma es una de las herramientas de diseño colaborativo más utilizada actualmente [44] (ver capítulo 2). En relación con nuestra propuesta, cuenta con una aplicación móvil de “mirror”<sup>7</sup>, disponible en las tiendas de los dispositivos, que permite visualizar un diseño ajustando las resoluciones al mismo teléfono. Figma permite extender su funcionalidad a través del desarrollo de plugins [45]. Existen a su vez una gran cantidad de plugins publicados por la comunidad que pueden ser instalados de manera gratuita. Para poder desarrollar un plugin es necesario crear una cuenta e instalar la aplicación desktop de Figma. Los plugins deben programarse utilizando HTML, CSS y JavaScript. En particular, la documentación utiliza TypeScript.

Algunas consideraciones sobre el desarrollo de plugins:

1. Lo más importante es el archivo **manifest.json**, el cual describe dónde se encuentran los archivos .js y .html (opcional) que son llamados al ejecutar el plugin.

---

<sup>7</sup> Aplicación que refleja en el celular lo que se está mostrando en la computadora. Estas aplicaciones requieren que se esté visualizando el diseño en la computadora para poder mostrarlo en el celular.

2. La forma de debugging es similar a un desarrollo web, interactuando con una consola del navegador.
3. Las llamadas a la API del navegador solo pueden hacerse desde el HTML (con un tag script por ejemplo). Este HTML se ejecuta dentro de un iframe que genera Figma y se visualiza en un modal o ventana emergente. Es posible ocultar esta ventana, y de esta forma ejecutar tareas en segundo plano como peticiones a un servicio externo. Desde un plugin, el desarrollador puede manualmente mostrar y esconder este modal como respuesta a algunos eventos que Figma posee.

## Limitaciones

- Desde la app móvil no se permite interactuar con los plugins, simplemente se puede ver un Frame o componente que tiene que estar seleccionado en la app Desktop (si no está seleccionado no se visualiza). Esta es la principal limitación, ya que nuestro plugin nunca es ejecutado desde la aplicación móvil.
- Durante el diseño Figma permite agregar “reacciones” a un componente, es decir comportamiento que permite cambiar de componente o diseño al hacer clic sobre algún elemento. Pero no es posible crear reacciones nuevas o asociarlas a componentes del diseño desde los plugins.
- Los elementos que la herramienta muestra son “dibujados” dentro de una etiqueta canvas [46], es decir que no son elementos HTML que puedan manejar eventos propios del lenguaje como click, hover, etc.

## Prueba de concepto

Dentro de las pruebas realizadas, se destacan dos estrategias que podrían haber funcionado y los inconvenientes que presentaron:

- Utilizando los eventos que brinda Figma [47], se intentó visualizar el modal cada vez que el usuario cambia el componente que tiene seleccionado, y dentro de dicho modal visualizar un botón que al hacerle clic, realiza el llamado a la API del navegador dentro un de una etiqueta **<script>**. Esta prueba no arrojaba errores en la consola del navegador, pero tampoco podía verse la ventana ni el botón desde la aplicación del teléfono.
- La segunda prueba era algo similar a la primera, pero configurando la ventana como oculta y haciendo directamente el llamado a la API del navegador para vibrar. En este caso, si nos encontramos con un error, el cual se puede visualizar en la figura 14. Este error ocurre debido a que los navegadores bloquean los llamados a esta función para que los programas no ejecuten vibraciones sin que el usuario lo desee, o sin que esté interactuando con el sitio.

```
✖ [Intervention] Blocked call to navigator.vibrate inside a cross-origin iframe because the frame has never been activated by the user: https://www.chromestatus.com/feature/5682658461876224. VM554:7
```

**Figura 14:** Error en segunda parte de la prueba de concepto con Figma.

## Adobe XD

Adobe XD es un editor de gráficos vectoriales desarrollado y publicado por Adobe Inc. Con esta herramienta se puede diseñar y crear prototipos de experiencia de usuario para aplicaciones web y móviles [48] (ver Capítulo 2).

De la misma forma que con Figma, es necesario crearse una cuenta y descargar la aplicación desktop que la empresa provee. También posee el concepto de plugin como pequeñas extensiones instalables que pueden ser creadas para expandir o generar nuevas funcionalidades, y también deben ser programados utilizando HTML, CSS y JavaScript [49].

Al instalar la aplicación, se crea en el sistema una carpeta llamada develop, donde debe ubicarse el código de los plugins.

## Limitaciones

El código de los plugins es ejecutado dentro de un motor similar al de los navegadores web [50], pero que no posee todas las funcionalidades de un navegador convencional. Por ejemplo, permite acceder a la API document para manipular los elementos y a la API fetch y XMLHttpRequest para hacer peticiones HTTP, pero no permite acceder al llamado a la API de vibración.

## Prueba de concepto

La prueba consistió en ejecutar el llamado a la API de vibración cuando el Plugin comenzaba a ejecutarse, pero esto arrojaba un error indicando que no existía dicha función.

## Sketch

Sketch es una herramienta de diseño que solo está disponible en plataformas de Apple (Mac y iOS) [51] (ver Capítulo 2). Posee también una aplicación móvil, que solo está disponible para teléfonos Iphone.

Sketch también permite la creación de plugins [52]. El código debe ser escrito en Javascript, lo cual podría haber significado una limitación ya que por ejemplo Safari no implementa la API Vibrate, como veremos más adelante. Sin embargo, Sketch provee una forma de hacer llamados a la API de OSX como si se estuviera programando en Objective-c. Esto lo hace a través de una librería que funciona de bridge [53], que permite

importar módulos o frameworks nativos y luego deja disponibles clases para invocarlos desde JavaScript.

Provee una herramienta de línea de comandos llamada SPKM [54], la cual facilita la publicación y prueba de plugins. Es necesario instalar la aplicación desktop para el desarrollo de plugins. La forma de debugging que la herramienta proporciona es a través de la visualización de logs.

Para poder ejecutar una vibración en el teléfono, es necesario utilizar el framework AudioToolbox [55]–[57], el cual es usado para reproducir sonidos. Esta herramienta nos provee una función que puede recibir un conjunto de sonidos predefinidos, y que también puede recibir una constante específica que indica que el dispositivo debe vibrar en lugar de reproducir sonido. La figura 15 ilustra las líneas de código necesarias para llevar a cabo una vibración.

```
__mocha__.loadFrameworkWithName('AudioToolbox');  
AudioServicesPlayAlertSound(kSystemSoundID_Vibrate);
```

**Figura 15:** Líneas de código necesarias para ejecutar una vibración en Sketch.

## Limitaciones

La forma de realizar llamadas a los frameworks o funcionalidades nativas del teléfono no está muy bien documentada, y es realmente difícil conseguir soporte o ejemplos de comportamiento similar a las pruebas que se estuvieron realizando.

## Prueba de concepto

Se generó un plugin que simplemente al cambiar el elemento seleccionado ejecute la función provista por el AudioToolbox framework para vibrar en el dispositivo móvil. Cuando esta función es ejecutada, en los logs de la herramienta se visualiza el error que se puede ver en la figura 16.

No fue posible obtener más información sobre el error dentro de la aplicación desktop.

También se realizó una prueba cambiando el llamado para vibrar por un simple mensaje, y este nunca se visualizó en la aplicación móvil, con lo cual llegamos a la conclusión que los plugins tampoco son ejecutados allí.

```
AudioServices.cpp:960:PlaySound: Failure to setup sound, err = -50
```

**Figura 16:** Error visualizado con prueba de concepto de Sketch.

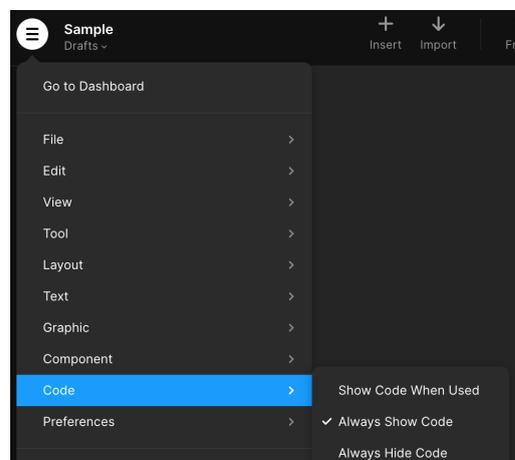
## Framer

Framer [58] es una herramienta de diseño y prototipado al igual que las anteriores mencionadas (más detalles en Capítulo 2), pero tiene la particularidad de que adicionalmente, se pueden crear componentes puramente con código, utilizando React [59]. Los diseñadores requieren tener conocimientos técnicos básicos para poder crear este tipo de componentes, pero sin embargo pueden no utilizar esta función y simplemente usar componentes existentes, o los publicados por la comunidad que son muy sencillos de personalizar.

En Framer no es posible crear plugins o extensiones de funcionalidad, pero sí se pueden publicar componentes de código, visuales o colores. Esto puede hacerse de manera pública para que cualquier usuario tenga acceso, o incluso restringir por equipos o de manera privada (sólo disponible en la versión paga)[60].

Framer posee una aplicación móvil, disponible desde las tiendas de los teléfonos tanto para Android como iOS, y también una versión desktop paga [61]. Para comenzar las pruebas solo fue necesario crearse una cuenta, ya que se puede interactuar con la herramienta desde cualquier navegador o descargando la aplicación de ordenador.

La funcionalidad de escribir componentes visuales utilizando código, debe ser activada por el usuario [62], ya que por defecto está deshabilitada. La forma de activar esta funcionalidad se encuentra ilustrada en la figura 17.



**Figura 17:** Activar componentes hechos con código en Framer.

Además de componentes completos, Framer permite definir Overrides [63], que son funciones JavaScript que se ejecutan en el momento que el componente se renderiza, es decir, cuando se visualiza el prototipo. Esta funcionalidad permite sobrescribir propiedades de componentes existentes publicados por otras personas o empresas, cambiando el comportamiento de éstos ante determinadas acciones. Por ejemplo, si nuestro componente recibía en sus props un manejador para la función **onClick**, con un Override podríamos, sin cambiar este componente, sobrescribir el manejador para este evento.

## Limitaciones

La posibilidad de utilizar librerías externas está solo disponible en la versión paga.

## Prueba de concepto

Se realizaron pruebas utilizando componentes completos y también overrides. Ambas pruebas fueron exitosas. Desde la aplicación móvil de los teléfonos Android, podemos visualizar los componentes del diseño y al interactuar con ellos, se genera la vibración esperada.

La primera prueba consistió en definir un Override que sobrescriba la propiedad `onClick` de un botón prediseñado que la herramienta provee. Este override simplemente hace un llamado a la API de vibración del navegador con un número fijo, con el simple efecto de poder percibir la vibración. En la figura 18 se puede ver el código del Override definido.

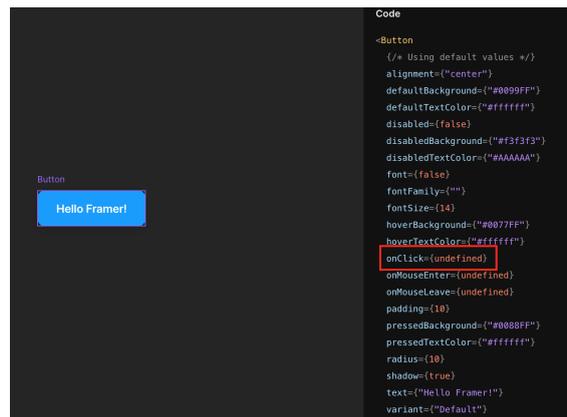
```
import { Override, Data } from "framer"

// [1]
const data = Data({
  pattern: 200,
})

// [2]
export function Vibrate(): Override {
  return {
    onClick: () => {
      window.navigator.vibrate(data.pattern)
    },
  }
}
```

**Figura 18:** Componente Override definido como parte de la prueba de concepto realizada con Framer.

Es importante destacar que el componente que utilicemos desde nuestro Override, debe exponer la propiedad que nosotros deseamos sobrescribir, y permitirnos hacerlo. La figura 19 muestra cómo verificar que el elemento que queremos utilizar con nuestro Override reciba la propiedad que deseamos sobrescribir.



**Figura 19:** Botón predefinido en Framer que recibe la propiedad `OnClick` y por lo tanto puede ser sobrescrita.

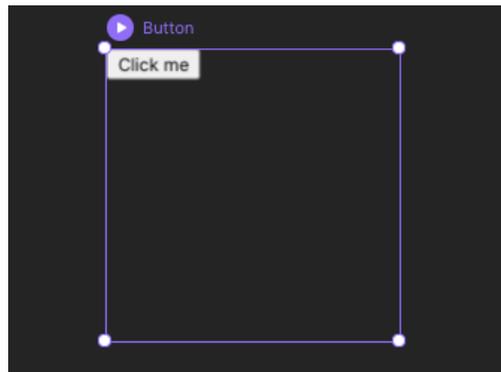
Sin embargo esta opción cuenta con ciertas limitaciones, como la posibilidad de recibir nuevos campos que no estaban definidos previamente en el componente. En nuestro caso, esta limitación no nos permitiría que el usuario pueda elegir la referencia a la vibración proveniente de nuestra aplicación, o elegir sobre qué evento quiere que se aplique el efecto (`onClick`, `onHover`, etc) [64].

La segunda prueba consistió en crear un nuevo componente utilizando código, el cual era simplemente un botón con el manejador del evento clic haciendo la llamada a la API de vibración. La figura 20 muestra el código del componente definido en esta prueba.

```
1 import * as React from "react"
2
3 export function Button(props) {
4   return (
5     <button
6       onClick={() => {
7         window.navigator.vibrate(200)
8       }}
9     >
10    Click me
11    </button>
12  )
13 }
```

**Figura 20:** Componente hecho con código utilizado como segunda prueba de concepto con Framer.

Luego, desde la interfaz web podíamos visualizar el botón creado e interactuar con él. Lo mismo sucedía desde la aplicación del teléfono, y al hacer clic en el botón, se ejecuta una vibración. En la figura 21 se muestra una captura de cómo se ve el componente desde dicha interfaz.



**Figura 21:** Componente de código definido en Framer, visto desde la interfaz web.

Luego de que las pruebas hayan sido satisfactorias con la herramienta Framer, decidimos utilizarla para como plataforma que nos permite probar las vibraciones y asociarlas a componentes visuales de cualquier diseño.

## Conclusión

Como resultado de la investigación realizada sobre las herramientas comerciales de prototipado de interfaces web-móviles y luego de realizar pruebas en ellas, se decidió utilizar la herramienta Framer.

Framer no solo posee una aplicación móvil que nos permite reproducir las vibraciones en el dispositivo, sino que también permite publicar paquetes que pueden ser utilizados por la comunidad. De esta forma podemos proveer más facilidades aún para los diseñadores ya que no necesitan generar componentes de código, ni tener conocimientos técnicos de programación para poder reproducir sus efectos hápticos. Además, es una herramienta de prototipado utilizada por muchos diseñadores, y cuenta con una gran comunidad que brinda soporte.

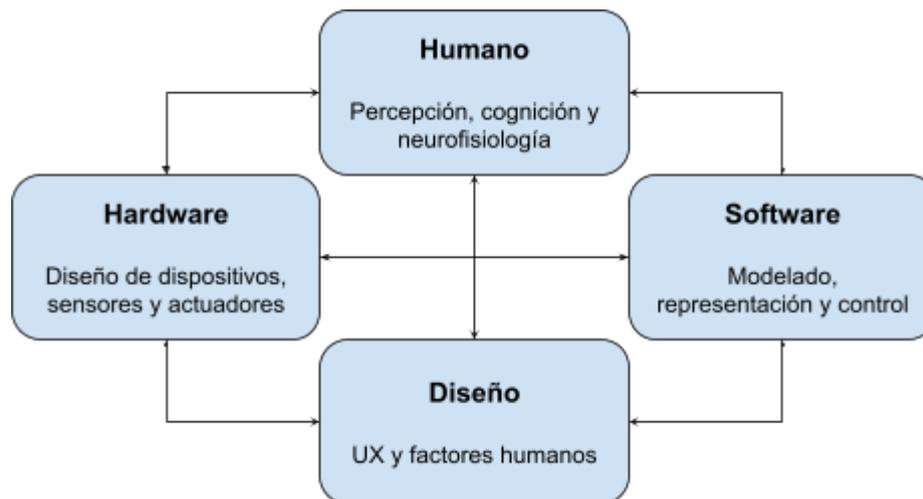
Si bien la herramienta posee algunas limitaciones dentro de su capa gratuita, consideramos que a los fines de este trabajo es una buena elección y permitirá a los diseñadores las facilidades que estamos buscando.

## 3.4 Tecnologías para la interacción háptica en móviles

La mayoría de los dispositivos móviles modernos incluyen hardware de vibración, que permite al software proporcionar información al usuario haciendo que el dispositivo se mueva o vibre. En esta sección se hará un análisis de las tecnologías y herramientas necesarias para poder llevar a cabo la ejecución de vibraciones en un teléfono.

## El stack háptico [65]

La implementación de la tecnología háptica es una experiencia de múltiples capas que requiere que la solución del sistema completo funcione en armonía para producir la mejor experiencia. Cada una de estas capas (Diseño, Software, y Hardware) es altamente dependiente de la otra. Por eso es nombrado “the haptic technology stack”. En la Figura 22, se presenta a modo ilustrativo la conexión entre las distintas capas y su interrelación.



**Figura 22:** Ilustración de háptica como un campo multidisciplinario

## Capa de Diseño [66]

La capa de diseño es donde todo se une para crear una experiencia háptica completa. Requiere un conocimiento profundo de varios enfoques de diseño, efectos hápticos y diseños de interacción. La capa de diseño es la que da contexto a la presente tesina y es posiblemente la más importante desde la perspectiva de la experiencia general, pero lo que es posible aquí puede verse limitado por las elecciones realizadas en las capas de software y hardware.

Esta capa está compuesta por:

- Guías de diseño
- Lenguaje háptico
- Herramientas de creación

Existen diferentes guías y lenguajes de diseño. Por ejemplo, la empresa Immersion ha propuesto unas guías que capturan casos de uso háptico [65] y sirven para definir cuál es el objetivo primario de una experiencia háptica en determinado producto, lo cual

luego facilita la elección del hardware y software subyacente. A continuación se explicarán los casos de uso capturados por esas guías.

## Cambio de atención

Los efectos hápticos de cambio de atención se utilizan para atraer la atención del usuario del sistema, indicando por ejemplo que algo está por ocurrir en su dispositivo.

## Momentos encantadores

Este tipo de experiencia háptica tiene como objetivo generar una sonrisa o un efecto positivo en los usuarios y suele asociarse con algún tipo de sorpresa. Buscan que el destinatario se sienta más conectado con el remitente. Por lo general, estos efectos se combinan estrechamente con gráficos y audio para proporcionar una mayor experiencia general.

## Flujo de información

Efectos hápticos que transmiten información al usuario sobre el estado (y cambios de estado) de la interfaz. Esto puede incluir la confirmación de la pulsación de un botón, el final de la lista u otro cambio de estado de la interfaz de usuario. Estos efectos también pueden ser una imitación de una acción real, como el clic de un botón.

## Inmersión sensorial

En las experiencias de juegos y realidad virtual, agregar efectos táctiles que resultan de las interacciones del usuario con el mundo virtual es una forma conocida de aumentar drásticamente la sensación de inmersión. Requiere una estrecha sincronización con el audio y vídeo y, a menudo, emplea efectos hápticos dinámicos, que responden al gesto de un usuario.

## Hedónicos

Hedonics es la generación de una respuesta emocional en los usuarios. La retroalimentación háptica es capaz de facilitar respuestas emocionales tanto positivas como negativas y el diseño de efectos es un mecanismo clave para ello. Por ejemplo, los efectos que tienen envolventes [67] suaves y están bien sincronizados con la experiencia general tienen más probabilidades de crear un efecto positivo, mientras que los efectos con envolventes nítidas tienen más probabilidades de generar un efecto negativo

## Ilusión

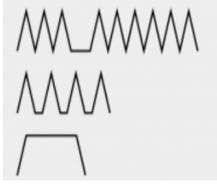
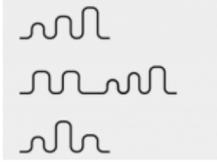
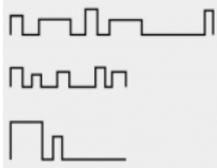
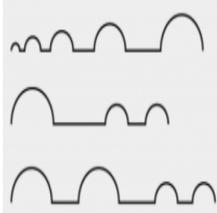
Este tipo de efectos buscan convencer al usuario que la experiencia táctil es completamente realista. El iPhone 6s es un gran ejemplo de esto: Apple pudo reemplazar un interruptor mecánico (el botón de inicio) en el teléfono con una sensación táctil y la mayoría de los usuarios no se dieron cuenta. Crear interruptores mecánicos realistas, texturas de superficie y otras sensaciones táctiles del mundo real es extremadamente desafiante y requiere el más alto nivel de ingeniería háptica.

## Sistema de codificación de efectos

Dado un caso de uso, asociar un significado con los efectos hápticos es un desafío. No hay gramática u otra construcción simbólica asociada con ellos y además hay muy pocas sensaciones hápticas en las que se acuerda comúnmente, por lo que es muy difícil lograr que los usuarios, en general, puedan extraer significado de una sensación háptica. Además, el significado de una sensación háptica suele depender mucho del contexto.

Los efectos hápticos simbólicos directos requieren una cantidad significativa de aprendizaje por parte de los usuarios finales, lo que puede significar una barrera demasiado grande para los dispositivos de consumo masivo.

Para lidiar con esta situación, Immersion propuso un sistema de codificación de efectos hápticos dependiente de la situación para dispositivos portátiles. Este sistema se encuentra descrito en la Tabla 2.

Categoría	Atención	Contexto	Ejemplos	Ritmo	Acción del usuario
Ahora esto	Interrumpir	No	Llamada telefónica		Tomar una acción con el dispositivo
			Temporizador		
			Alarma		
Revisar esto	Ambiente	No	SMS		Tomar una decisión de qué hacer.
			Mensaje de Voz		
			Correo electrónico		
Saber esto	Interrumpir	Si	Recordatorio		Tomar conocimiento de información importante
			Logro alcanzado		
			Batería baja		
Hacer esto	Ambiente	Si	Acelerar		Realizar una acción con el cuerpo
			Intervalo de correr		
			Girar a la izquierda		
Cambiar esto	Enfocar	Si	Cambiar de modo		Asegurarse de que su cambio surgió efecto
			Seguimiento activado/desactivado		
			Cambio de deporte		

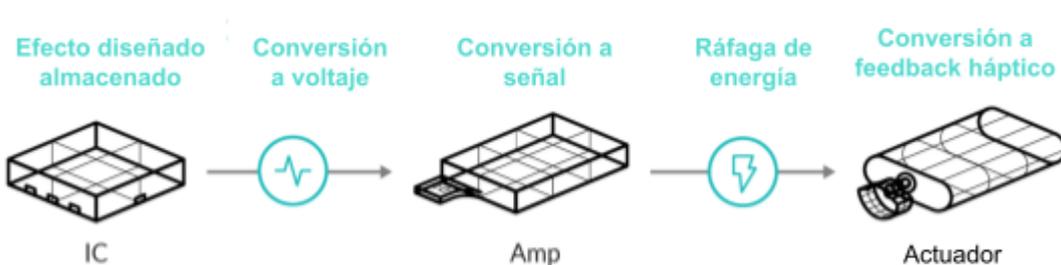
**Tabla 2:** Sistema de codificación propuesto por Immersion [66].

## Capa de Software [68]

La capa de software interactúa tanto con la electrónica del controlador, como con las aplicaciones que utilizan háptica. Es la encargada de orquestar el hardware háptico mediante la generación de señales en respuesta al estado de la aplicación o las interacciones del usuario. Esta capa está compuesta por:

- Códec
- API
- Sistemas operativos

El flujo planteado para poder representar las señales es ilustrado en la figura 23.



**Figura 23:** Flujo de ejecución de una vibración, hasta que llega al actuador.

Es común que los circuitos integrados de controladores hápticos proporcionen algún tipo de búfer u otros comandos en el almacenamiento en el chip, que se pueden activar y que darán como resultado la generación de un voltaje. Los datos almacenados en estos búferes normalmente están muy cerca de la señal que se envía al amplificador y siempre son específicos del controlador del IC. Esta capacidad se usa generalmente para productos hápticos que no tienen un sistema operativo, ya que el búfer se puede activar directamente con una señal I2C u otros comandos de bajo nivel. Esto se utiliza generalmente para efectos similares a botones por su muy baja latencia y dependencia limitada del estado del software.

Si bien las capacidades de los efectos hápticos dependen en gran medida de la elección del hardware, la actualización a un mejor hardware háptico no necesariamente equivale a más capacidades. Con un actuador de mayor graduación y un controlador IC de mayor rendimiento, la capa de software juega un papel más importante en el aprovechamiento de las capacidades del hardware háptico.

Los efectos hápticos también pueden ser representados en series de tiempo con una frecuencia de muestreo regular. Los sistemas operativos como Android e iOS pueden digerir búferes de dichos datos de señal, generalmente con una frecuencia de muestreo de 1 ms, y representarlos en el hardware háptico disponible. Con este mecanismo de codificación, la experiencia háptica que se generará variará ampliamente en la práctica debido a la variación mecánica a nivel motor.

Las APIs<sup>8</sup> disponibles por los fabricantes de dispositivos poseen enfoques muy variados, lo que puede representar una barrera para desarrolladores que deberán implementar o testear su aplicación en cada plataforma y modelo de hardware distinto. A continuación presentaremos algunos ejemplos de estas APIs.

## Core Haptics (IOS)

Lanzado a mediados del año 2019. Utiliza un modelo de abstracción que permite especificar los efectos mediante envolventes acústicas con fases ADSR [67]. Estos envolventes se analizan en tiempo de ejecución y se utilizan para sintetizar una señal específica del motor, que luego es enviado al amplificador, y luego al actuador.

## Android Vibrate (Google)

Android no admite la abstracción de efectos y, en cambio, sólo proporciona una interfaz de señal de bajo nivel, que se representa como un búfer de amplitudes de 8 bits que se muestrean<sup>9</sup> a 1 ms [69].

## Open XR 1.0 (Khronos)

Esta API está dirigida a casos de uso de XR y juegos, y está destinada a proporcionar una abstracción de hardware consistente y de bajo nivel para aplicaciones de alto rendimiento [70]. La codificación de efectos en OpenXR es una lista de efectos de vibración individuales, cada uno de los cuales tiene una duración, frecuencia y amplitud fijas.

Conceptualmente, esta API es un poco más sofisticada que la API de Android. Aún así, no abstrae de manera significativa la estrecha dependencia del hardware y es probable que sufra inconsistencias si se reproducen las mismas listas de efectos en dispositivos con diferentes actuadores.

## Capa de Hardware [71]

Los actuadores se encargan de convertir las señales eléctricas en estimulación táctil del usuario del dispositivo. Es análogo a una pantalla para el sistema de visión humana. Esta capa está compuesta por:

- Actuador

---

<sup>8</sup> Acrónimo para “Application Programming Interface”. Software que permite a dos aplicaciones comunicarse entre sí.

<sup>9</sup> El proceso de muestreo consiste en tomar muestras de una señal a intervalos regulares de tiempo.

- Controlador IC

## Actuadores y mecanorreceptores

El objetivo de un actuador háptico es estimular el sistema de mecanorreceptores descritos en el Capítulo 2 para crear una experiencia táctil. En la Tabla 3 se pueden visualizar los distintos tipos de actuadores que se utilizan y el tipo de estímulo que generan sobre los mecanorreceptores involucrados.

Tipo de actuador	Tipo de estimulación	Mecanorreceptor relevante
Electromecánico	Vibración	Pacinian
Piezoeléctrico	Vibración + Textura	Pacinian
Kinestésico	Deformación + Fuerza	Merkel + Meissner
Superficie	Fricción	Meissner + Pacinian
Otro	Estiramiento + Deformación	Ruffini + Meissner + Merkel

**Tabla 3:** Actuadores, estímulos y mecanorreceptores [71].

En los dispositivos móviles, foco de esta tesina, se utilizan actuadores electromagnéticos. De las cuatro subcategorías de este tipo de actuadores (de masa excéntrica, de resonancia lineal, de bobina o solenoides), los fabricantes de celulares han optado por los siguientes dos:

- ERM o Motores Rotativos con Masa Excéntrica. Motores de rotación con una masa montada en el eje de forma descentrada. Cuando se aplica voltaje a estos motores, vibran mientras la masa descentrada gira. Son buenos para sensaciones de tipo retumbante de baja frecuencia, pero son bastante lentos, por lo que no son útiles para generar efectos hápticos precisos. Progresivamente, estos motores han sido reemplazados por la siguiente subcategoría.
- LRA o Actuadores por Resonancia Lineal. Actuadores lineales que consisten de una masa en un resorte y un electroimán. El electroimán se carga y descarga, lo que hace que la masa vibre a una frecuencia específica. Son extremadamente eficientes en términos de energía, y esto los convierte en

una excelente opción para teléfonos móviles y otros dispositivos que funcionan con baterías.

## Consideraciones eléctricas

Cada una de las soluciones descritas anteriormente debe tener un controlador IC o amplificador que pueda convertir las señales de efecto háptico en señales que puedan comprender los actuadores. En la mayoría de los casos, la solución eléctrica la dicta el actuador, ya que cada uno requiere un tipo específico de señal de accionamiento. En la Tabla 4 se presentan algunas consideraciones, analizando también algunos de los distintos tipos de actuadores.

Voltaje	El voltaje que pueden generar los controladores.  Para actuadores electromagnéticos, muchos dispositivos utilizan 10V o más, mientras que para actuadores piezoeléctricos, se encuentran en un rango de entre 60-120V
Rango de frecuencia	Frecuencia de salida que pueden generar los controladores.  Para los actuadores de tipo LRA, el controlador necesita rastrear la frecuencia de resonancia.
Velocidad de subida	Algunas experiencias hápticas requieren una rápida transición entre no voltaje al voltaje más alto. Para alcanzar este tipo de respuesta, los controladores deben ser capaces de cambiar los niveles de salida rápido.

**Tabla 4:** Consideración eléctrica de los controladores y actuadores [71].

## API de vibración W3C

Además de las APIs provistas por los fabricantes de dispositivos que hemos repasado en la sección anterior, el Consorcio World Wide Web (W3C) presentó una API de vibraciones para aplicaciones web.

El W3C es una comunidad internacional que propone y desarrolla estándares web [72]. Uno de esos estándares creados por este consorcio es la definición de la API de vibraciones [73].

Como pudimos notar, la mayoría de las herramientas de diseño analizadas permiten generar plugins o diseñar utilizando código con JavaScript, en un entorno muy similar al de

cualquier navegador Web. Existe una gran cantidad de navegadores web que cuentan con una implementación de esta API, que nos permite pedirle al motor de vibración del dispositivo (si es que éste posee) [43], que ejecute un patrón de vibración. Debido a que con este trabajo se intenta generar una herramienta web para poder realizar el diseño de vibraciones, se eligió esta API como herramienta para la generación de las vibraciones en los dispositivos.

La API de vibraciones se destaca principalmente por su facilidad de uso. A través del llamado a la función **vibrate** se puede iniciar una vibración del hardware pasando como argumento un sólo número, o un arreglo de un sólo número. Este número representa la cantidad de milisegundos que el motor de vibración estará encendido. La figura 24 muestra la instrucción necesaria para hacer un simple llamado a esta API, produciendo como resultado una pequeña vibración.

```
window.navigator.vibrate(200);  
window.navigator.vibrate([200]);
```

**Figura 24:** Demostración de llamado a la API de vibraciones que producirá una vibración de 200 milisegundos.

Para la creación de vibraciones más complejas se puede utilizar un arreglo de números. Este arreglo describe un patrón que alterna entre secuencias de encendido y apagado del motor. Las posiciones impares del arreglo representan la cantidad de milisegundos que el motor se encontrará prendido, mientras que las posiciones pares representan la cantidad de milisegundos que el motor estará apagado. Por ejemplo, el llamado que se muestra en la figura causaría que el dispositivo vibre por 200 milisegundos, luego se detenga por 100 milisegundos y comience a vibrar nuevamente por otros 200 milisegundos. Este ejemplo se ve ilustrado en la figura 25.

```
window.navigator.vibrate([200, 100, 200]);
```

**Figura 25:** Llamado a la API de vibraciones, apagando y prendiendo el motor del dispositivo.

No existe una limitación en la cantidad de vibraciones/pausas que se pueden especificar, y el arreglo puede tener un tamaño par o impar. No importa que el último valor del arreglo sea una pausa, ya que el celular dejará de vibrar de todas formas al final de cada vibración.

Sin embargo, esta API posee varias limitaciones:

## Compatibilidad

Una de las mayores limitaciones está en su compatibilidad entre los distintos navegadores. Como se puede observar en la imagen, los llamados a esta API no están disponibles desde Safari o iOS Safari, con lo cual las vibraciones solo pueden ser probadas desde dispositivos Android. La figura 26 muestra una versión detallada de la compatibilidad de esta API en distintas plataformas.

	🖥️						📱				
	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	WebView Android	Chrome Android	Firefox Android	Opera Android	iOS Safari
vibrate	32	79	16 ★	No	No	No	4.4.3 ★	32 ★	79 ★	Yes ★	No

**Figura 26:** Compatibilidad de la API de vibraciones en distintas plataformas, según especifica el sitio de Mozilla [43].

A su vez, se realizaron algunas pruebas en un dispositivo Samsung Galaxy J7 y no se pudieron reproducir las vibraciones. Existen dispositivos o marcas que realizan su propia implementación o adaptación de los navegadores web, con lo que la presencia o ausencia de esta API puede variar.

## Control sobre el motor de vibración del dispositivo

Esta API solo nos permite interactuar con los motores de los dispositivos enviándoles señales de encendido y apagado. Es decir, solo podemos indicar cuánto tiempo queremos que el dispositivo vibre, y/o cuánto tiempo queremos que esté apagado. No es posible otro tipo de personalización del patrón, como podría ser por ejemplo atenuar su intensidad.

Además tampoco tiene soporte para manipular más de un actuador del dispositivo, si es que este lo tuviese.

## 3.5 Arquitectura de la solución

Esta sección pretende documentar las decisiones de diseño y arquitectura de la solución utilizadas en este trabajo, justificando y detallando los caminos elegidos hasta llegar al armado de la solución completa. Además, mencionar varias de las técnicas y conceptos asociados al procesamiento de señales que fueron necesarios para la construcción de la herramienta.

Como se describió en capítulos anteriores, utilizaremos Framer como herramienta de prototipado, ya que nos permite reproducir las vibraciones en los dispositivos móviles y acoplar esto a diseños de interfaz existentes. Los diseñadores podrán agregar comportamiento vibratorio a cualquier elemento de su diseño, como respuesta a una acción del usuario.

Para que los diseñadores puedan diseñar sus vibraciones, se creó una herramienta web que permite el diseño y customización de vibraciones. Luego se creó un componente en Framer (el cual fue publicado) al que se le puede proporcionar una URL de la herramienta y de esta forma reproducir la vibración diseñada.

La herramienta web creada está compuesta por dos aplicaciones. Una backend, que contiene la lógica de almacenamiento y manipulación de la información, y una frontend que es la encargada de la visualización de los datos y gráficos. A continuación detallaremos cada una de las partes que componen la herramienta, tecnologías utilizadas y consideraciones técnicas. Finalizamos esta sección con una demostración del flujo completo necesario para reproducir los escenarios planteados en la sección [Escenarios de uso](#).

## Backend

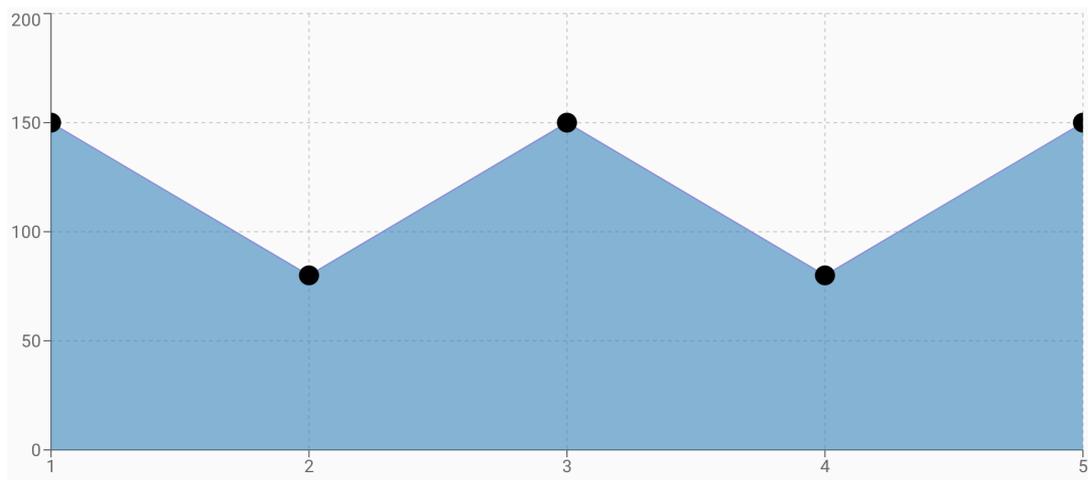
Una de las partes que componen la herramienta desarrollada es una aplicación backend. Esta aplicación consta de un API Rest que provee un conjunto de endpoints a través de los cuales es posible almacenar, modificar u obtener las vibraciones.

Esta aplicación es un componente totalmente aislado, independiente de Framer y de la aplicación Frontend. Sin embargo, tanto Framer como la otra aplicación se comunican con este servicio backend para poder obtener los datos almacenados y mostrarlos (o ejecutar las vibraciones en el caso de Framer).

La decisión de crear una aplicación independiente fue tomada por una cuestión de simplicidad, ya que permitía trabajar en el procesamiento de la información sin interferir o cegar por decisiones de diseño de los otros componentes de la herramienta. Además, de esta forma fue posible crear una aplicación que no depende de ninguno de los otros servicios, e incluso podrían crearse distintas aplicaciones frontend que consumas de sus endpoints, o extender este trabajo para aplicaciones nativas en dispositivos móviles, o utilizar cualquier otra herramienta de diseño en lugar de Framer.

## Generación del patrón de efectos

Permitir diseñar una vibración no es tan trivial. Para poder proveer esta funcionalidad se ideó una forma de poder expresar en un gráfico como debería sentirse el efecto deseado. De esta forma, el usuario podrá editar los puntos representados en el gráfico para generar o modificar el efecto tanto como desee. La figura 27 muestra dicho gráfico con algunos puntos en él.



**Figura 27:** Gráfico para diseño de vibraciones en herramienta web.

Los valores expresados en el gráfico deben poder reproducir un efecto particular en un dispositivo móvil en el marco del escaso control que nos provee la API de vibraciones del navegador sobre el motor de vibración de los teléfonos. Otro factor a tener en cuenta es la duración de la vibración, la cual debería ser configurable por los diseñadores que podrían necesitar diseñar distintos efectos para diferentes escenarios. Por ejemplo, un diseñador podría querer generar una vibración para una notificación mensaje o de llamada entrante, donde sabemos que ésta última puede perdurar más en el tiempo ya que requiere que el usuario conteste en el momento, con lo cual es posible que el efecto que se desea generar dure más que una simple notificación de un mensaje que el usuario del dispositivo puede ver más tarde.

## Muestreo

Como podemos observar en la imagen de la sección 3.3.1, el usuario configura una serie de puntos en un gráfico, es decir genera una señal que representa cómo la vibración debe comportarse. Las señales son patrones de variaciones que representan o codifican información. Esta señal es un ejemplo de una señal digital. Aunque en este caso particular no podemos escribir una ecuación simple que describa la gráfica, el gráfico en sí puede tomarse como una definición de la función que asigna un número  $s(t)$  a cada instante de tiempo (cada valor de  $t$ ) [74].

A los fines de poder enviar los datos al motor de vibración de los dispositivos móviles, debemos obtener una representación analógica de la señal. Esto se puede hacer muestreando una señal en puntos aislados, igualmente espaciados en el tiempo. El resultado es una secuencia de números que se puede representar como una función de una variable de índice que sólo toma valores enteros.

Para el desarrollo de esta herramienta, se utiliza una cantidad fija de muestras por cada par de puntos (se utilizó un valor de 10 muestras). El algoritmo consiste en recorrer todos los puntos que graficó el usuario, comparando cada uno con su siguiente (si es que

éste tiene). Se calcula la diferencia entre estos dos puntos, y se divide esa cantidad por la constante con la cantidad de muestras. Luego, dependiendo la pendiente de la gráfica dada por la comparación entre el valor de estos puntos, se calculan los valores intermedios para el muestreo. Para esclarecer este algoritmo, presentamos un pseudocódigo a modo ilustrativo de la comparación entre cada par de puntos.

```
si punto1 = punto2
  devolver 10 elementos iguales
si punto1 > punto2
  diferencia = (punto1 - punto2) / 10
si punto1 < punto2
  diferencia = (punto2 - punto1) / 10
fin

resultado = []

// El primer valor es punto1 y se realizan sumas o restas
dependiendo

// si el gráfico es creciente o decreciente.
iterador = punto1
repetir 10 veces
  resultado.agregar(aux)
  si punto1 > punto2
    iterador = iterador + diferencia
  sino
    iterador = iterador - diferencia
  fin
fin

devolver resultado
```

## PWM

La modulación por ancho de pulso (PWM por sus siglas en inglés Pulse Width Modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica, ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

El ciclo de trabajo describe la proporción de tiempo "encendido" dentro de un intervalo o "período" de tiempo. Un ciclo de trabajo bajo significa baja potencia, porque la alimentación está apagada la mayor parte del tiempo.

El ciclo de trabajo se expresa en porcentaje, donde el 100% significa que el motor está encendido durante todo el período de tiempo. Cuando una señal digital está encendida

la mitad del tiempo y apagada la otra mitad del tiempo, la señal digital tiene un ciclo de trabajo del 50%. Cuando una señal digital pasa más tiempo en el estado encendido que en el estado apagado, tiene un ciclo de trabajo mayor al 50%, y cuando pasa más tiempo en el estado apagado que en el estado encendido, tiene un ciclo de trabajo menor al 50% [75].

En el contexto de este trabajo, esta técnica fue utilizada para poder simular la intensidad de las vibraciones. Debido a las limitaciones de la API de vibraciones, y la imposibilidad de controlar otra cosa que no sean el prendido y apagado del motor de vibraciones de los dispositivos móviles, con el uso de ésta técnica fue posible generar un efecto más cercano a la vibración que el usuario espera con valores más altos o más bajos en la gráfica.

Se establecieron 5 posibles valores del ciclo de trabajo, que representan los porcentajes en los que el motor se encontrará prendido:

- El valor 0 indica que el motor estará apagado durante todo el ciclo de trabajo.
- El valor 0.25 indica que el motor estará prendido el 25% del tiempo y se encontrará apagado el restante 75%.
- El valor 0.5 indica que el motor estará prendido la mitad del tiempo y apagado la otra mitad.
- El valor 0.75 indica que el motor estará prendido el 75% del tiempo y apagado el 25% del tiempo restante.
- El valor 1 indica que el motor estará prendido durante todo el ciclo de trabajo.

Basándonos en el muestreo obtenido como se explicó en la sección [Muestreo](#), utilizaremos esos elementos para generar un nuevo conjunto de números con el formato esperado por la API de vibraciones, indicando la cantidad de milisegundos que el motor debe encontrarse prendido y la cantidad en la que debe encontrarse apagado.

Teniendo en cuenta la duración de la vibración que el diseñador configuró, se convertirá ese valor en milisegundos. Luego se hará una división de la cantidad de elementos de la muestra con esa cantidad, y el resultado de ese cálculo representa la cantidad de milisegundos que cada elemento de la muestra puede utilizar, es decir la duración del ciclo de trabajo.

Cada elemento de la muestra tendrá asociado un valor del ciclo de trabajo de los 5 presentados anteriormente. Esta asignación estará dada por la comparación de cada elemento con su siguiente, de esta forma, si el valor siguiente es mayor al actual se asignará al siguiente valor de ciclo de trabajo, y si es menor se asignará el anterior, con excepción de los valores 0 y 1, que representan el mínimo y máximo respectivamente. Por ejemplo, si el elemento 1 tiene un valor de ciclo de trabajo de 0.5, y el elemento 2 es mayor a él, el elemento dos tendrá un valor de ciclo de trabajo de 0.75. Ahora bien, si el elemento 1 hubiese tenido asociado un valor del ciclo de trabajo de 1, el elemento 2 también tendría asociado ese mismo número ya que es el valor más grande.

Por cada elemento se generarán 2 nuevos valores. Por ejemplo, si para un elemento el ciclo de trabajo es 0.5 y la duración del ciclo de trabajo es 10ms, este elemento resultará en el par [5, 5].

## Tecnologías utilizadas

La aplicación backend fue desarrollada utilizando el framework JavaScript Express [76], que se caracteriza por su facilidad de uso y simplicidad. Posee un conjunto de endpoints que permiten crear nuevas vibraciones, editarlas, borrarlas y también obtener su representación como lo espera la API de vibraciones, utilizando las técnicas de procesamiento de señales descritas en la sección anterior.

Las vibraciones creadas son almacenadas en una base de datos no-SQL. Esta base de datos es provista por un servicio llamado Firestore [66], el cual forma parte de la plataforma de Google Firebase [77]. Si bien es un servicio pago, posee una capa gratuita bastante amplia, que permite el uso de varias de sus herramientas sin siquiera ingresar una tarjeta, y luego si se superan los límites es posible continuar usando los servicios pagando [78].

A su vez, esta aplicación fue hosteada utilizando la plataforma Heroku [79], que se integra muy fácilmente con Github y permite actualizar las aplicaciones cuando el repositorio es actualizado con nuevo código.

## Frontend

Llamamos frontend a la aplicación con la que los usuarios interactúan. Es la encargada de la visualización de los gráficos, junto con la lógica necesaria para poder crear, editar y borrar puntos en él. A su vez, debe encargarse de obtener la información almacenada en la base de datos, comunicándose con los endpoints que expone la aplicación backend. De esta forma, obtiene la información y es capaz de mostrarla a través de una interfaz gráfica. También debe comunicarse con el servicio backend para el envío de nueva información, cada vez que un usuario crea una nueva vibración, o se edita una existente.

Esta aplicación fue desarrollada utilizando el framework de desarrollo web ReactJS [80] y se encuentra publicada en Firebase hosting [77], el servicio de la plataforma Firebase que permite alojar aplicaciones estáticas.

## Framer

A continuación se describirán más en detalle algunas de las limitaciones mencionadas de la herramienta Framer.

## Limitaciones de la versión gratuita

Anteriormente hemos mencionado algunas de las limitaciones de la versión gratuita de la herramienta. En esta sección haremos un repaso sobre todas las limitaciones y explicaremos más en detalle aquellas que fueron relevantes para el desarrollo del trabajo.

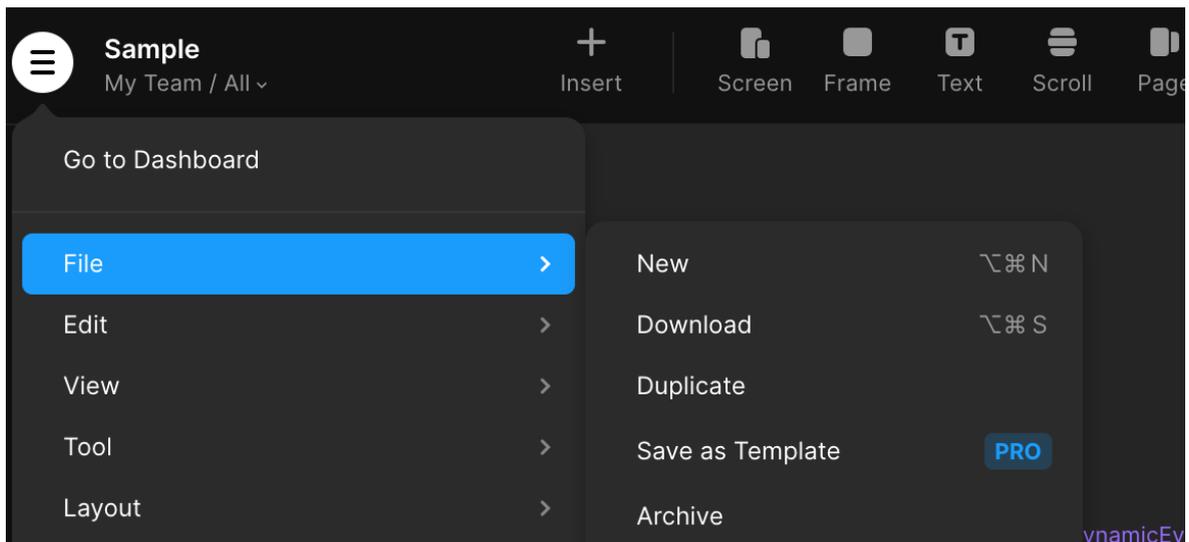
La herramienta provee dos tipos de cuenta: la versión gratuita y la versión **pro**. Existen algunas limitaciones relacionadas a la cantidad de proyectos que una cuenta gratuita puede tener, como también la cantidad de editores que pueden trabajar sobre el mismo proyecto [81]. Recientemente Framer publicó una nueva aplicación Desktop gratuita (la cual se encuentra en Beta), pero no presenta mayores ventajas o funciones que la aplicación web existente [82].

Las principales limitaciones están relacionadas con la posibilidad de utilizar la aplicación Desktop y de acceder a los archivos locales. La aplicación Desktop que provee la herramienta permite utilizar funciones tales como crear templates, proyectos privados y publicar paquetes compartidos por equipos. En particular una de las limitaciones más importantes que se encontraron fue a la hora de intentar publicar un **package** [60] y la imposibilidad de descargar los diseños en formato **framerfx**.

## Publicar un paquete

Al utilizar la aplicación de escritorio “legacy” de Framer, es posible generar una carpeta con el contenido de todo el proyecto [83] descargándolo en un formato **framerfx**, lo cual permite realizar un seguimiento de las versiones utilizando herramientas de versionado como Git, como también integraciones con Github o plataformas de CI/CD, y también a través del uso de la librería **framer-cli** [84] publicar versiones de paquetes tanto públicas como privadas y actualizarlas con nuevo contenido.

Luego de investigar y realizar varias pruebas, aún sabiendo que la documentación de Framer sugiere que para publicar paquetes es necesario tener la versión paga de la herramienta, fue posible publicar una versión de los componentes creados con código para poder generar vibraciones en el diseño. La versión web permite descargar los diseños, aunque no en un formato **framerfx** como la versión paga, sino en un formato **framerx**, el cual no puede convertirse en una estructura de carpetas ni ser modificado, su único uso es ser importado nuevamente. La figura 28 muestra la funcionalidad de descarga, accesible desde la aplicación web de Framer en su versión gratuita.



**Figura 28:** Descargar diseño desde herramienta web de Framer.

Utilizando el archivo descargado con formato framerx, y con el uso de la herramienta de línea de comandos framer-cli, fue posible publicar una versión pública de los componentes de código. La figura 29 muestra la salida del comando utilizado para publicar el paquete de manera exitosa.

```
[→ Framer git:(master) npx framer-cli publish Sample.framerx --public --new=VibrationComponents2
[Ready to publish @framer/v4szacvm.vibrationcomponents2 v1.0.0 (public store). Continue (y/N)? y
🚀 Package successfully published
```

**Figura 29:** Paquete publicado exitosamente en Framer a través de la herramienta de línea de comandos.

Adicionalmente nos encontramos con el problema de que estos componentes publicados no pueden ser actualizados, ya que obtenemos un error de que necesitamos algunos metadatos, que solo están disponibles al utilizar la estructura de carpetas del formato framerfx. Esta situación se encuentra ilustrada en la figura 30.

```
[→ Framer npx framer-cli publish Sample.framerx --public
❌ Cannot fetch package metadata because of missing information.
Please re-run this command with a display name using the --new argument.
```

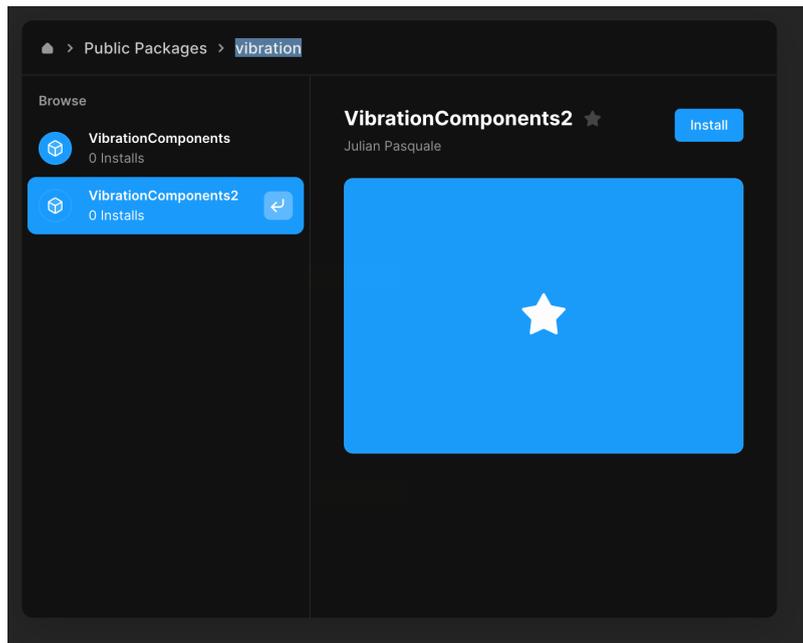
**Figura 30:** Error al intentar actualizar paquete publicado en Framer.

Al intentar con el mismo comando utilizado para publicar por primera vez los componentes (usando el flag --new), nos aparece que el package ya está publicado. Debido a esta limitación, se creó un segundo paquete con una versión final de los componentes, llamado **VibrationComponents2**. La figura 31 muestra dicho error.

```
[→] Framer npx framer-cli publish Sample.framerx --public --new=VibrationComponents
✖ This package has already been published as VibrationComponents (@framer/v4szacvm.vibrationcomponents).
Please re-run the publish command without the '--new' argument.
```

**Figura 31:** Error al intentar publicar un paquete que ya existe.

El paquete publicado puede ser usado instalándolo a través del buscador de paquetes que provee la herramienta. Al instalar el paquete, todos los componentes de código definidos pueden ser utilizados. En la figura 32 se muestra la búsqueda del paquete publicado en la herramienta web de Framer.



**Figura 32:** Paquete publicado disponible para instalación desde la herramienta web de Framer.

### 3.6 Demostración

En el campo de la Interacción Hombre Computadora el método más utilizado para validar una herramienta de diseño es la demostración [85]. Una demostración muestra lo que la herramienta puede soportar y cómo los usuarios pueden trabajar con ella. En su forma más básica consiste de ejemplos que muestran su expresividad, exhibiendo el umbral de acceso, el techo de posibilidades o el espacio de diseño. La demostración que se describe más abajo se basa en esta última forma para incluir una exploración inicial de las posibilidades de diseño con ejemplos desde diferentes puntos en ese espacio [86]. Una exploración del espacio de diseño ejemplifica la amplitud de las aplicaciones admitidas por la herramienta al encajarlo en un tema de investigación más amplio.

Esta demostración se estructura en dos partes. En la primera, utilizaremos las historias de usuario que presentaremos a continuación. Estas historias nos ayudarán a

presentar cada una de las funcionalidades que la herramienta provee, y luego se explicará en detalle cómo llevarlo a cabo.

En la segunda parte, presentamos una forma posible de integrar las herramientas que proponemos en el diseño y prototipado de dos aplicaciones, a partir del catálogo de ejemplos que provee Framer como material de capacitación y ejercitación a sus usuarios [87].

## Parte 1. Demostración con historias de usuario

Una historia de usuario es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario [88]. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos [89]. Son tareas de desarrollo que se suelen expresar como "persona + necesidad + propósito" y representan una descripción general e informal de una función de software.

En el marco de esta tesina, las historias de usuario serán utilizadas para documentar las distintas funcionalidades que la herramienta provee.

### Crear nueva vibración

<p><b>Como</b> diseñador UX,</p> <p><b>Quiero</b> crear una nueva vibración</p> <p><b>Para poder</b> agregar interacción háptica a mi diseño de interfaces.</p>
<p><b>Criterios de aceptación:</b></p> <p><b>Dado</b> un diseñador en la página principal del sistema</p> <p><b>Cuando</b> hace clic en el botón de crear nueva vibración,</p> <p><b>Entonces</b> se visualiza un modal donde deberá ingresar el nombre y la categoría de la vibración que desea crear.</p>

### Visualizar vibración

<p><b>Como</b> diseñador UX,</p> <p><b>Quiero</b> visualizar los datos de una vibración creada</p> <p><b>Para poder</b> determinar si requiere algún tipo de modificación, o si los parámetros son acordes a mi necesidad.</p>
--

**Criterios de aceptación:**

**Dado** un diseñador en la página principal del sistema

**Cuando** busca la vibración que desea visualizar en la barra lateral izquierda, y hace clic en su nombre,

**Entonces** lo redireccionará a la pantalla de edición de vibraciones, donde podrá tanto ver como editar los parámetros de dicha vibración.

## Borrar vibración

**Como** diseñador UX,

**Quiero** borrar una vibración del sistema

**Para poder** remover la vibración de la interfaz y evitar confusiones al crear nuevas.

**Criterios de aceptación:**

**Dado** un diseñador en la página de edición de vibraciones

**Cuando** hace clic en el botón Borrar Vibración,

**Entonces** el sistema borrará la vibración, haciendo que ya no sea posible acceder a ella.

## Editar vibración

**Como** diseñador UX,

**Quiero** editar una vibración

**Para poder** ajustar o perfeccionar el efecto al escenario de uso dentro de un sistema.

**Criterios de aceptación:**

**Dado** un diseñador en la página de edición de vibraciones

**Cuando** edita, borra o agrega algún punto de los que se visualizan en el gráfico representado a la vibración,

**Entonces** el gráfico de la pantalla se ajustará reflejando el nuevo estado de la vibración.

## Compartir una vibración

<p><b>Como</b> diseñador UX,</p> <p><b>Quiero</b> compartir una vibración creada</p> <p><b>Para poder</b> consultar la opinión de colegas y compartir el desarrollo con la comunidad.</p>
<p><b>Criterios de aceptación:</b></p> <p><b>Dado</b> un diseñador en la página de edición de vibraciones</p> <p><b>Cuando</b> hace clic en el botón para copiar link de vibración,</p> <p><b>Entonces</b> el enlace es copiado a su portapapeles para poder ser compartido.</p>

## Actualizar efecto luego de cambiar patrón de vibración

<p><b>Como</b> diseñador UX,</p> <p><b>Quiero</b> actualizar el efecto vibrotáctil en mi diseño, luego de haber realizado cambios desde la aplicación web</p> <p><b>Para poder</b> probar nuevamente la vibración y analizar si cumple los requerimientos de mi diseño.</p>
<p><b>Criterios de aceptación:</b></p> <p><b>Dado</b> un diseñador luego de haber realizado una modificación en una vibración desde la aplicación web, y de tener dicha vibración configurada en su diseño</p> <p><b>Cuando</b> sacude su teléfono Android desde la aplicación de Framer,</p> <p><b>Entonces</b> la aplicación de Framer le muestra un menú con un botón que dice "Refrescar", que al ser apretado actualiza los datos de la vibración.</p>

## Implementación de historias

Luego de la presentación de las historias de usuario, explicaremos a continuación cómo llevar a cabo cada una de ellas a través de la inclusión de imágenes y ejemplos prácticos concretos.

## Crear nueva vibración

Ingresar al sitio web (<https://tesis-99ba6.web.app/>) de la aplicación. Allí se podrá observar un botón para crear una nueva vibración, ubicado en la parte superior izquierda de la pantalla. Este botón se muestra en la figura 33.



**Figura 33:** Captura del botón como se muestra en el sitio web de la herramienta.

Al hacer clic sobre ese botón, aparecerá una ventana emergente (o modal) solicitando un nombre y categoría para la vibración a crear. Ambos campos son de texto libre, cualquier palabra o conjunto de caracteres puede ser ingresado. Esta ventana se verá de la misma forma que lo ilustra la figura 34.

Una ventana emergente con el título 'Nueva Vibración'. Contiene dos campos de texto: 'Nombre \*' con el valor 'Vibración 1' y 'Categoría \*' con el valor 'Categoría 1'. En la parte inferior hay dos botones: 'CANCELAR' (gris) y 'GUARDAR' (azul).

**Figura 34:** Ventana emergente utilizada para crear nuevas vibraciones.

Luego de llenar ambos campos y hacer clic en el botón Guardar, se creará la vibración en el sistema. La nueva vibración fue creada con el nombre y categoría provistos, pero se utiliza una configuración por defecto para el resto de los valores. Esta configuración por defecto establece que la primera vez, la vibración tendrá una duración de 1 segundo, junto con un patrón de 5 puntos dibujados en el gráfico que representan una línea recta, simulando un efecto vibrotáctil constante.

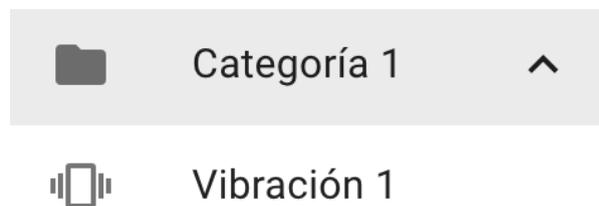
## Visualizar vibración

Siguiendo con el ejemplo de la vibración creada en la historia anterior, esta nueva vibración podrá ser visualizada dentro de una carpeta con el nombre de la categoría ingresada, en la barra lateral izquierda de la pantalla. La figura 35 muestra dicha carpeta colapsada, y luego de hacer clic en ella se abrirá mostrando las vibraciones agrupadas bajo dicha categoría permitiéndonos acceder a cada una, como lo ilustra la figura 36. Además, si

había otras carpetas abiertas, estarán cerradas solo permitiendo ver las vibraciones de la última carpeta elegida.



**Figura 35:** Demostración de carpeta colapsada con el nombre de la categoría creada, utilizada para agrupar todas las vibraciones bajo la misma categoría.



**Figura 36:** Demostración de carpeta sin colapsar con el nombre de la categoría creada, utilizada para agrupar todas las vibraciones bajo la misma categoría.

Luego de hacer clic sobre la vibración que se desea visualizar, el sistema lo redirigirá a una nueva pantalla donde podrá ver toda la información de la vibración elegida. A esta pantalla la llamaremos “pantalla de edición de vibraciones”. Esta pantalla se encuentra ilustrada en la figura 37.



**Figura 37:** Pantalla de edición de vibraciones.

## Borrar vibración

El borrado de una vibración se puede realizar desde la pantalla de edición de vibraciones. En el costado superior derecho, luego del nombre de la vibración, puede observarse un botón con el texto “Borrar vibración”. Al hacer clic en dicho botón, la vibración será borrada del sistema. Este botón puede verse en la figura 38.



**Figura 38:** Botón para borrar una vibración.

## Editar vibración

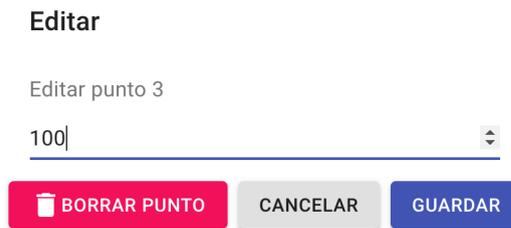
Para poder editar una vibración, debemos primero localizarla en el panel lateral izquierdo, dentro de la categoría o carpeta correspondiente, y luego acceder a la pantalla de edición de vibraciones. En esta pantalla veremos un gráfico con los valores que la vibración tiene configurados hasta el momento, junto con algunos botones y acciones que podemos realizar.

De cada vibración podemos modificar tanto su patrón como su duración. El resto de los valores como el nombre o categoría no son modificables.

Existen múltiples formas en las que un diseñador puede adaptar el patrón a su necesidad, y el sistema permite llevar a cabo 3 acciones para esto:

- Cambiar o adaptar un punto existente.
- Eliminar un punto del gráfico.
- Crear nuevos puntos en el gráfico.

Para poder editar un punto existente, se debe hacer clic sobre el punto en cuestión, haciendo que el sistema muestre una ventana emergente con el valor actual del punto, y permitiéndonos cambiarlo por uno nuevo, entonces el diseñador puede realizar la modificación y clicar el botón guardar para que ésta sea llevada a cabo. Esta ventana emergente se encuentra ilustrada en la figura 39, y será la misma que utilizaremos para borrar puntos.



**Figura 39:** Ventana emergente utilizada para editar el patrón de las vibraciones.

El procedimiento para poder remover un punto del gráfico es exactamente el mismo, con la excepción de que no debemos completar el campo con un nuevo valor, sino que debemos directamente presionar el botón **Borrar punto** de la ventana emergente ilustrada en la figura 39.

En caso de querer agregar un nuevo punto al patrón, se debe hacer clic en el botón Agregar nuevo punto que se visualiza en la parte inferior derecha de la pantalla. La figura 40 ilustra dicho botón.



**Figura 40:** Botón para agregar un nuevo punto.

Al hacer clic sobre dicho botón, el sistema mostrará una ventana emergente que permite ingresar el valor para este nuevo punto. Dicha ventana puede verse ilustrada en la figura 41.



**Figura 41:** Ventana emergente para añadir nuevos puntos al gráfico.

Finalmente, además del patrón de la vibración utilizando el gráfico, el diseñador puede personalizar la duración de la vibración. Para hacer esto debe completar con el campo que se visualiza en debajo del gráfico, en el lado izquierdo. Este campo debe contener la cantidad de segundos que se espera que dure la vibración, y está acompañado de un botón con un icono de guardado; el botón se visualizará con un color gris si la cantidad de segundos es la que la vibración tiene configurada, pero en caso de realizar un cambio sobre la duración, el botón tendrá un color azul, indicando que debe ser cliqueado

para efectuar la actualización en la duración. La figura 42 muestra el campo junto con el botón, cuando la cantidad de segundos ya fue actualizada, y la figura 43 muestra el mismo escenario pero cuando la cantidad de segundos aún no fue guardada.



**Figura 42:** Campo de duración de vibración actualizado, junto con botón para enviar



**Figura 43:** Campo de duración de vibración sin actualizar, junto con botón para enviar de color azul.

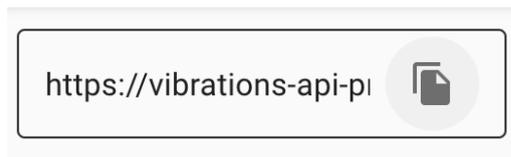
## Compartir una vibración

Podemos pensar en dos escenarios distintos en los que los diseñadores van a querer compartir sus vibraciones:

1. Un diseñador quiere compartir con otro un enlace al sistema para pedirle a algún colega que colabore con él en el diseño del efecto vibratorio.
2. Un diseñador quiere compartir con alguien un enlace que pueda utilizar para probar la vibración, y analizar si esa persona siente el efecto que el diseñador está buscando crear.

Para el primero de los casos, basta simplemente con copiar el enlace del sitio web. Al ser una página pública y que no requiere autenticación, cualquier persona puede visualizar o modificar las vibraciones del sistema solo con la URL de la vibración.

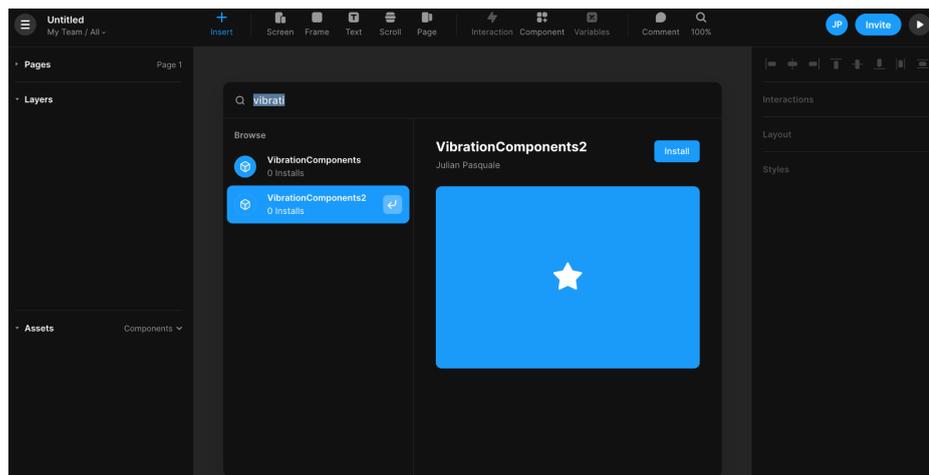
Para el segundo caso, donde el enlace del sistema no es suficiente ya que lo que se quiere no es el gráfico sino un enlace que permita obtener la versión “ejecutable” de la vibración (versión apta para la API de vibraciones), el sistema tiene una funcionalidad que permite copiar al portapapeles un enlace que nos permite obtener dicha representación. Para esto, se debe hacer clic en el botón de copiar, ubicado en el costado superior izquierdo de la pantalla de edición de vibraciones, como lo muestra la figura 44.



**Figura 44:** Botón para copiar enlace a vibraciones en formato ejecutable por la API.

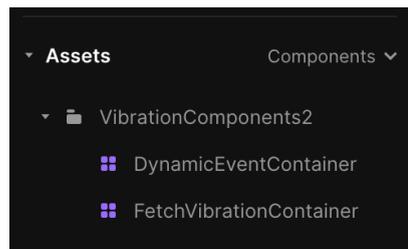
## Agregar vibración a mi diseño

Para agregar vibraciones a un diseño realizado en Framer, es posible utilizar los paquetes de la herramienta que fueron publicados en este trabajo, como se explicó en la sección [3.5.3.2 Publicar un paquete](#). Para hacer uso de estos paquetes, se deben instalar desde el menú para insertar elementos, dentro del diseño. Al hacer clic en el botón Insertar que se puede ver en la barra superior de Framer, se abre una ventana emergente que nos permite buscar por los paquetes creados. Esta acción se ilustra en la figura 45.



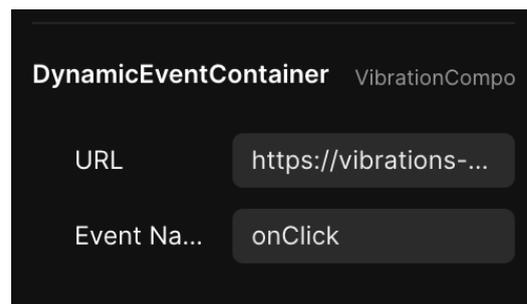
**Figura 45:** Instalando paquete de vibraciones en diseño de Framer.

Luego de instalar el paquete, tendremos disponibles todos los elementos que éste posea y los podremos usar en nuestro diseño. Tendremos acceso a los componentes instalados a través de la barra lateral izquierda, donde en la parte inferior veremos una carpeta con el nombre del paquete que dentro contiene todos los componentes que pueden ser arrastrados para agregarlos al diseño. Esto se ve ilustrado en la figura 46.



**Figura 46:** Componentes disponibles luego de instalar paquetes.

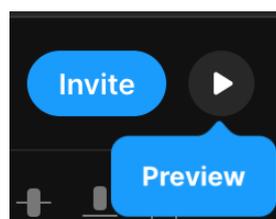
Al instalar los paquetes provistos y agregarlos a nuestro diseño, veremos que éstos nos permiten pasarles un enlace. En dicho campo agregaremos la URL que nos permite copiar la herramienta web. Al usar el componente **DynamicEventContainer** veremos que además nos pide ingresar un evento. Este evento indica cuándo se disparará la vibración, y por defecto tiene el valor **onClick**. Esta configuración de parámetros debe hacerse desde la barra lateral derecha dentro de la herramienta Framer, luego de agregar el componente al diseño y hacer clic en él. Podemos ver los campos configurables en la figura 47.



**Figura 47:** Configuración de componente con enlace a vibración y evento.

## Probar vibración

Luego de instalar los paquetes provistos y agregar el enlace de la vibración que deseamos probar, debemos utilizar la aplicación móvil de Framer desde nuestro dispositivo Android para poder probar el efecto creado. Para esto debemos activar el modo Preview del diseño desde la herramienta Web de Framer, utilizando el botón con un ícono de Play que se visualiza en el costado superior derecho de la pantalla, como lo muestra la figura 48.



### Figura 48: Botón para activar preview de diseño

Al activar el modo preview, podremos ver una ventana como la que se muestra en la figura 49. En dicha ventana veremos un botón que dice “Mobile”, que nos mostrará un código QR que puede ser escaneado desde el teléfono para abrir la vibración, o bien un botón que dice “Copy Link” que copiará el enlace a nuestro portapapeles, listo para ser enviado a otras personas o utilizado por nuestro propio dispositivo.

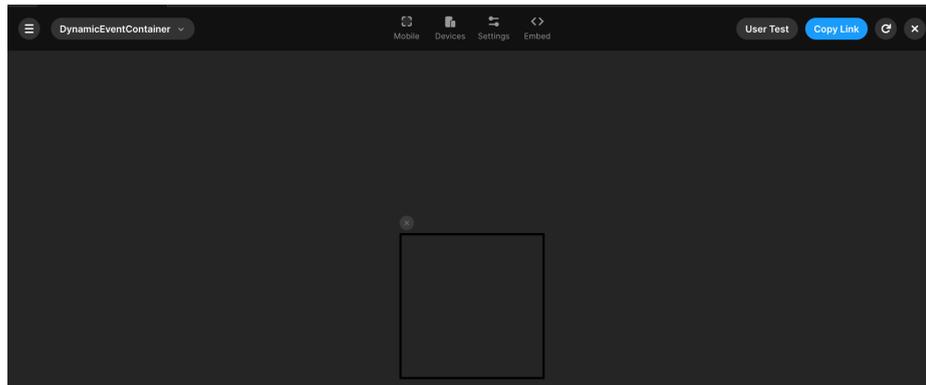


Figura 49: Framer con modo preview activado

## Actualizar efecto luego de cambiar patrón de vibración

Comúnmente los diseñadores buscarán realizar cambios en el patrón de vibración o en su duración con el objetivo de refinar dicho diseño, y luego necesitarán volver a probar el efecto para poder seguir personalizándolo.

Luego de haber probado la vibración en el teléfono y realizar cambios en dicha vibración, para poder probar nuevamente el patrón actualizado se debe sacudir el teléfono Android en la aplicación Framer, y de esta forma, aparecerá un menú como el que se muestra en la figura 50. Se debe seleccionar la opción “Refresh”, que actualizará el diseño junto a la vibración luego de haber sido modificada.

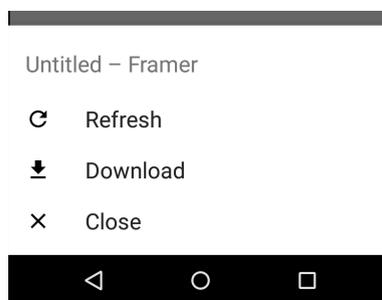


Figura 50: Menú para refrescar vibración en teléfono Android.

## Parte 2. Una forma de integrar los efectos vibrotáctiles durante el prototipado de aplicaciones

En esta segunda parte de la demostración mostraremos dos ejemplos prácticos de cómo agregar efectos vibrotáctiles en el proceso de diseño y prototipado de aplicaciones.

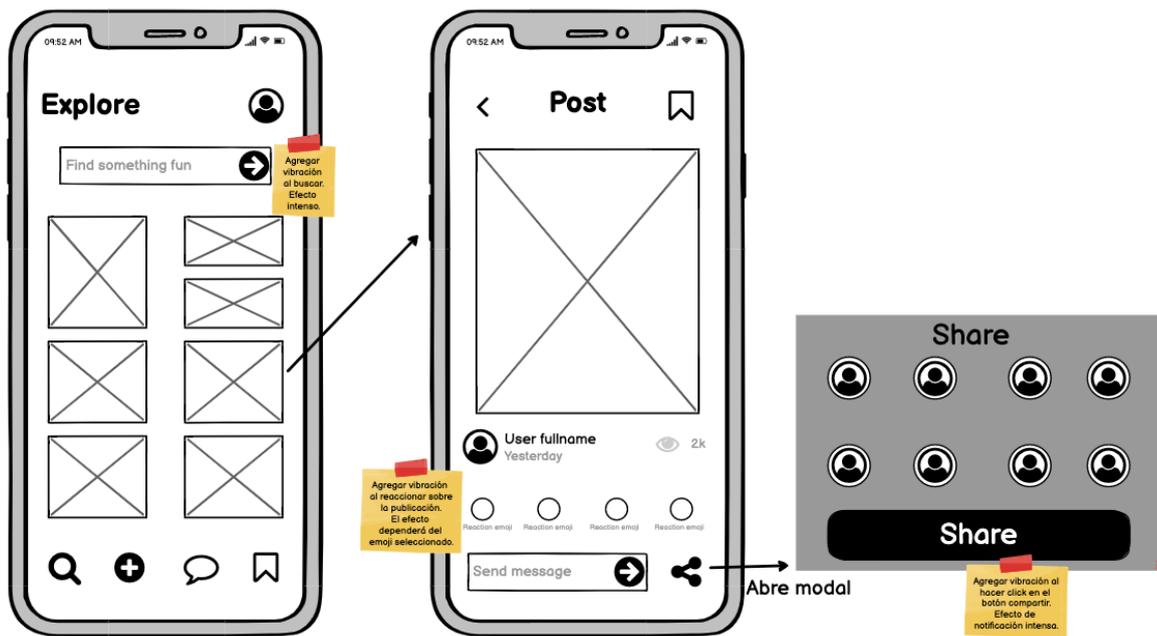
Para esto, utilizaremos ejemplos provistos por la misma herramienta Framer en su página web. En la url <https://www.framer.com/examples> Framer provee un amplio conjunto de ejemplos de prototipos detallados con aplicaciones de diferente tipo. El objetivo de este catálogo es doble, por un lado presenta ejemplos de logros posibles con la herramienta para el prototipado, por otro genera una especie de librería de patrones para el diseño.

Creemos que utilizar un par de ejemplos representativos de diferentes dominios de aplicación puede contribuir a presentar la potencia, utilidad y usabilidad de nuestra propuesta durante el diseño y prototipado de UX que integre efectos hápticos vibrotáctiles

### Social app

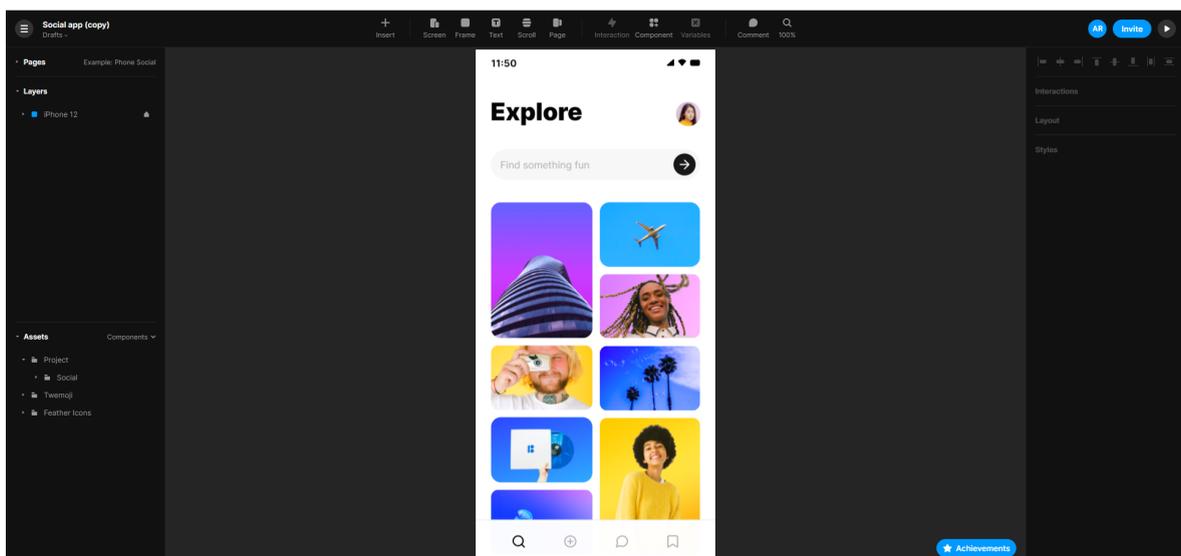
El primer ejemplo es el denominado “Social app”. El objetivo de este tutorial es crear interacciones basadas en una galería de imágenes. Al modo de Instagram, presenta un conjunto de pantallas e interacciones que simulan ser una aplicación de fotos donde una persona puede ver las publicaciones de otros usuarios. También provee funcionalidades para compartir una publicaciones con otros usuarios, o hacer un filtrado sobre el listado de publicaciones.

Siguiendo un proceso habitual, un diseñador podría haber llegado a un boceto de bajo nivel en el que indica a modo de storyboard la secuencia de interacciones posibles que llevan de una pantalla a la siguiente, como se muestra en la Figura 51, en las herramientas actuales (aquí presentado el caso de Balsamiq [37]), para registrar dónde quiere efectos de vibración y de qué tipo, podría agregar comentarios vinculados a cada elemento de pantalla.



**Figura 51:** Prototipo de baja fidelidad de “Social app”.

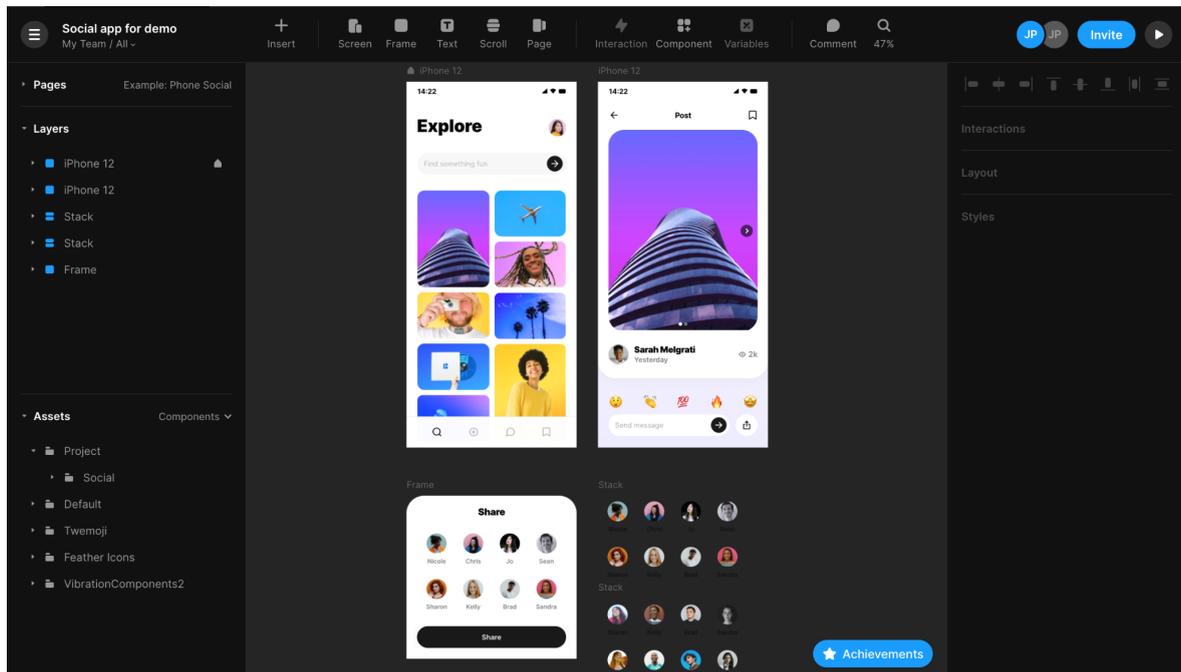
Una vez conseguida la validación del esquema interactivo propuesto, el siguiente paso es aumentar la fidelidad del prototipo para luego pasar al desarrollo. Aquí entra en juego una herramienta como Framer, la cual es utilizada por el diseñador para generar un prototipo de la primera pantalla, como se muestra en la figura 52.



**Figura 52:** Pantalla principal del ejemplo Social App [90].

Luego de esta primera pantalla, el diseñador generará un prototipo de las pantallas restantes (la pantalla del detalle de una publicación y el modal para compartirla), llegando

así al ejemplo completo que Framer provee en su sitio [87]. La figura 53 muestra el prototipo propuesto implementado en la herramienta Framer.



**Figura 53:** Prototipo de Social app implementado en Framer [90].

Supongamos que el diseñador, desea explorar la interacción vibrotáctil agregando tres tipos de vibraciones (de todos los efectos que estaban previstos en el diseño original). En particular, va a explorar ahora:

- Al hacer una búsqueda filtrando las publicaciones.
- Al hacer clic en el botón de compartir una publicación.
- Al reaccionar sobre una publicación con el emoticón de fuego.

Con nuestra propuesta, el primer paso es hacer clic sobre el prototipo de ejemplo en la página de Framer para poder generar una copia del diseño provisto, asegurándonos de comenzar desde el mismo diseño que el sitio oficial provee. Una vez hecho esto, es posible modificar nuestra propia copia del diseño y modificarlo como queramos.

Una vez generada la copia del diseño, el siguiente paso fue instalar la librería **VibrationComponents2**, creada como parte de este trabajo de tesina. Luego fue necesario generar los efectos que esperábamos que se reproduzcan ante cada uno de los eventos en la aplicación web para diseñar vibraciones [91]. Para esto generamos una nueva categoría que nos permite agrupar todas las vibraciones dentro de una misma carpeta, y luego creamos 3 vibraciones distintas en ella. Esto se puede ver ilustrado en la figura 54.

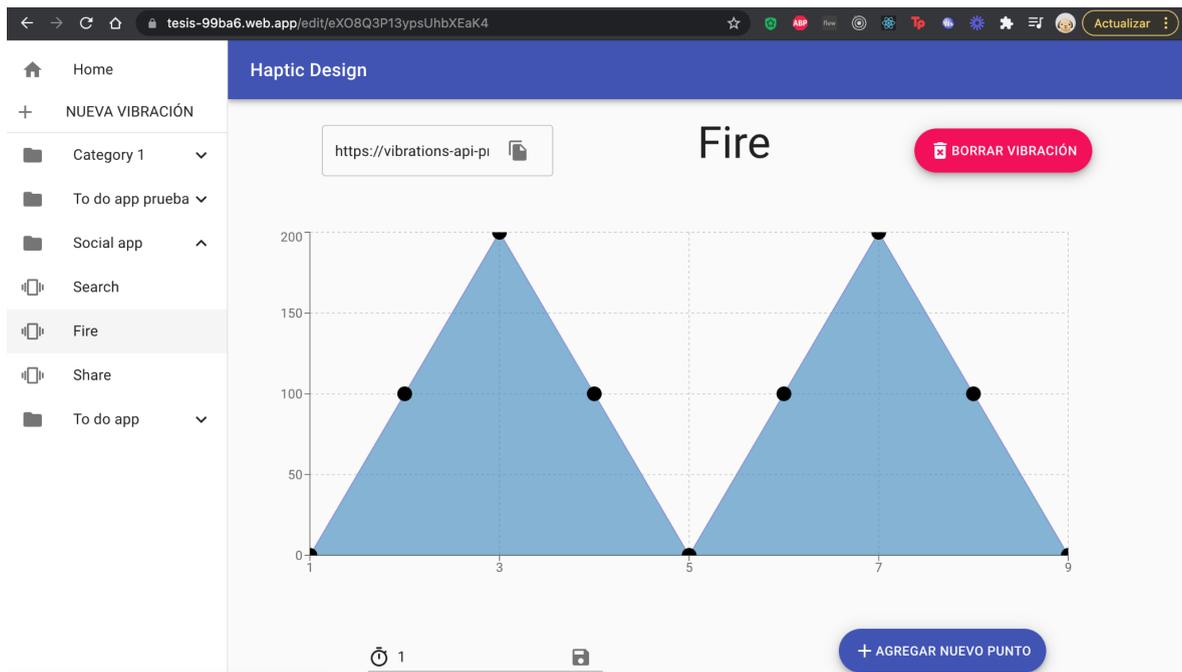


Figura 54: Vibraciones creadas para la aplicación Social app.

Para agregar la vibración al momento de realizar un filtrado sobre las publicaciones, solo bastó con colocar el componente **DynamicEventContainer** de la librería, sobre el botón de enviar presente en el diseño. De esta forma, al hacer clic sobre dicho botón el usuario podrá sentir el efecto creado. La figura 55 muestra cómo fue colocado el componente sobre el botón.

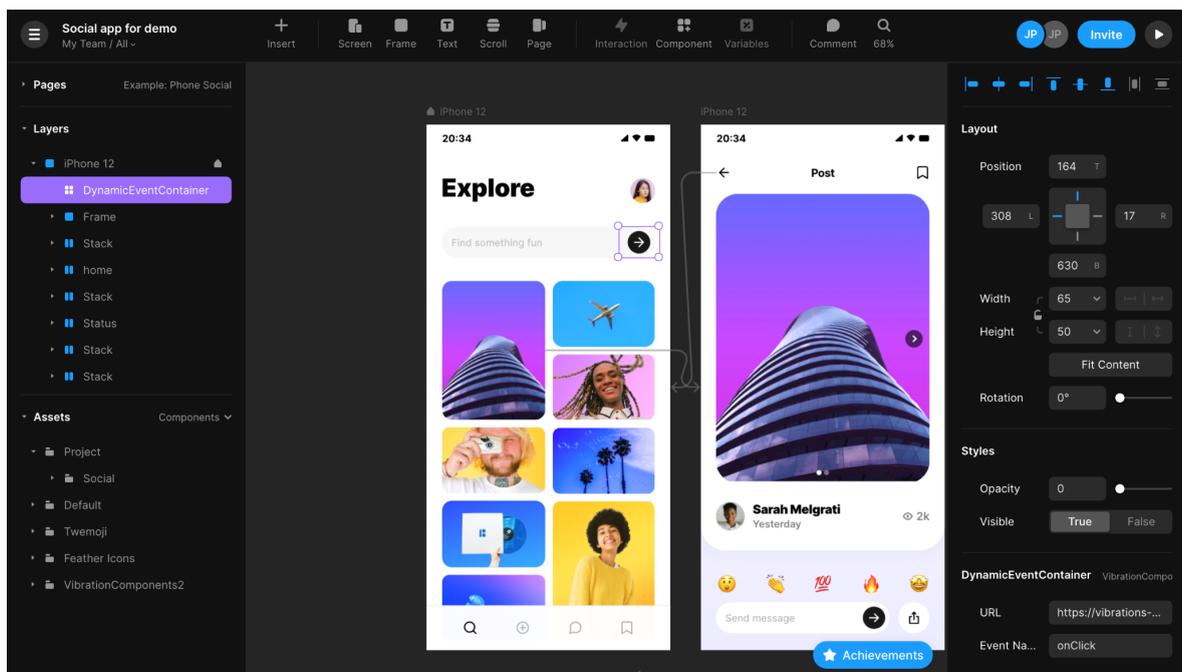
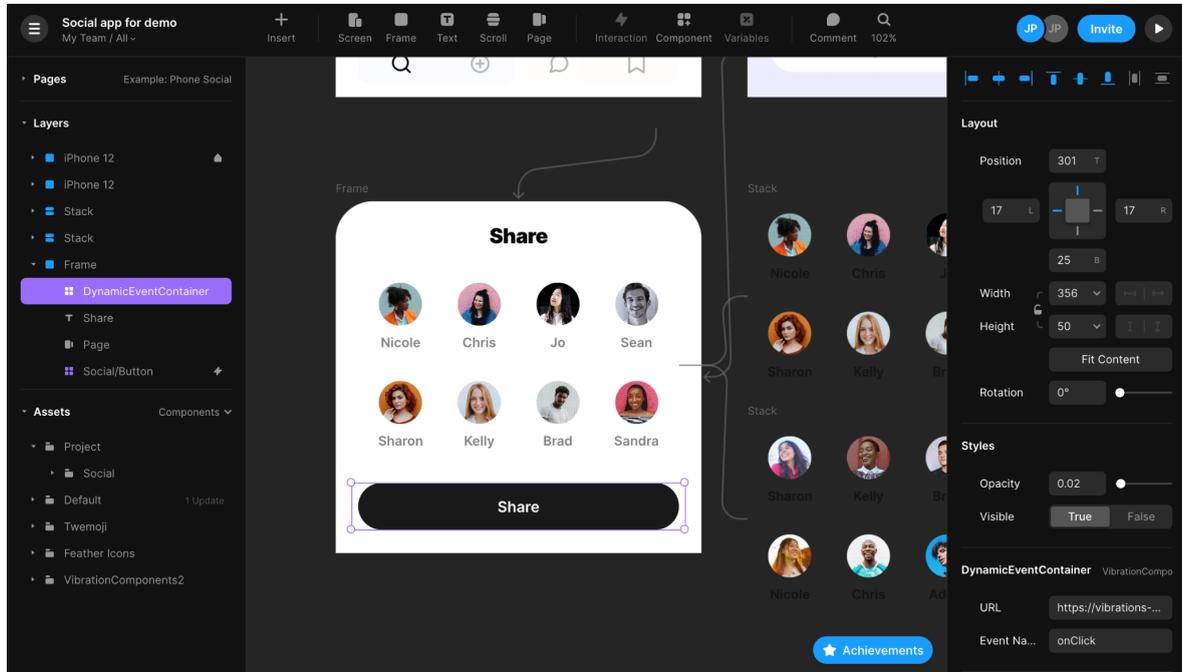


Figura 55: Componente para ejecutar vibración colocado sobre un botón.

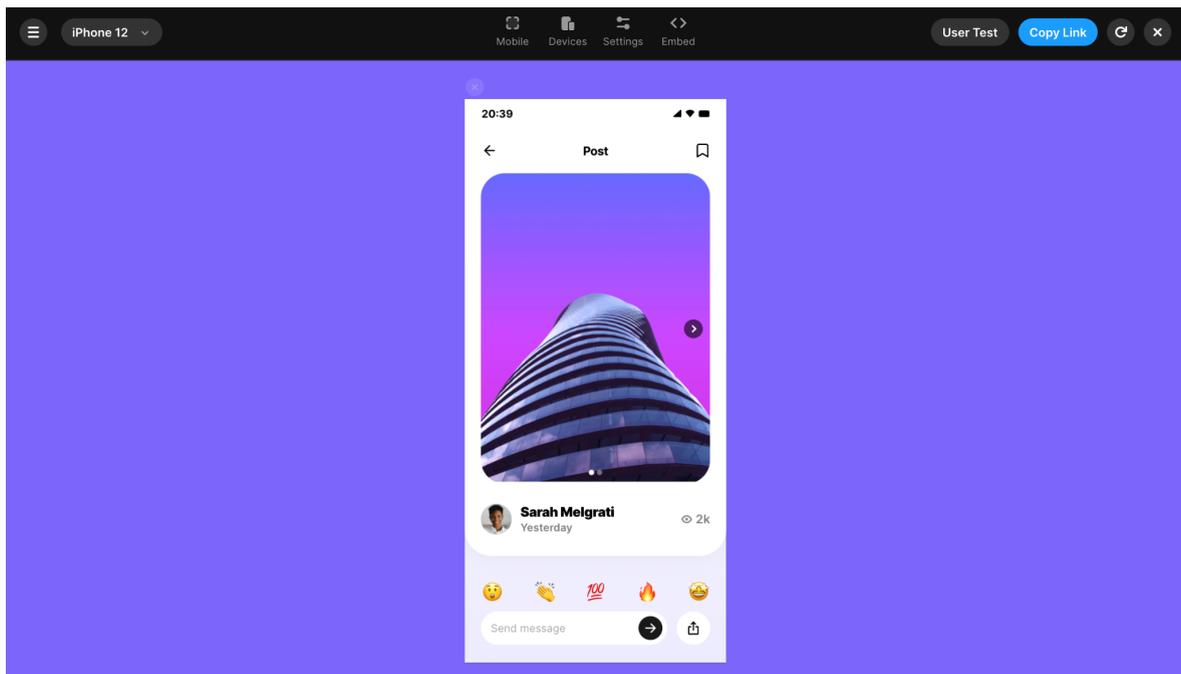
Luego el proceso para agregar una vibración al momento de compartir una publicación fue similar. En este caso dicha funcionalidad se encuentra ubicada en una ventana emergente que es accesible al abrir una publicación y hacer clic en el botón de compartir. En esta ventana se puede encontrar un botón con el texto “Share”, el cual fue utilizado para agregar el efecto. Dicho botón junto con el componente agregado se presentan en la figura 56.



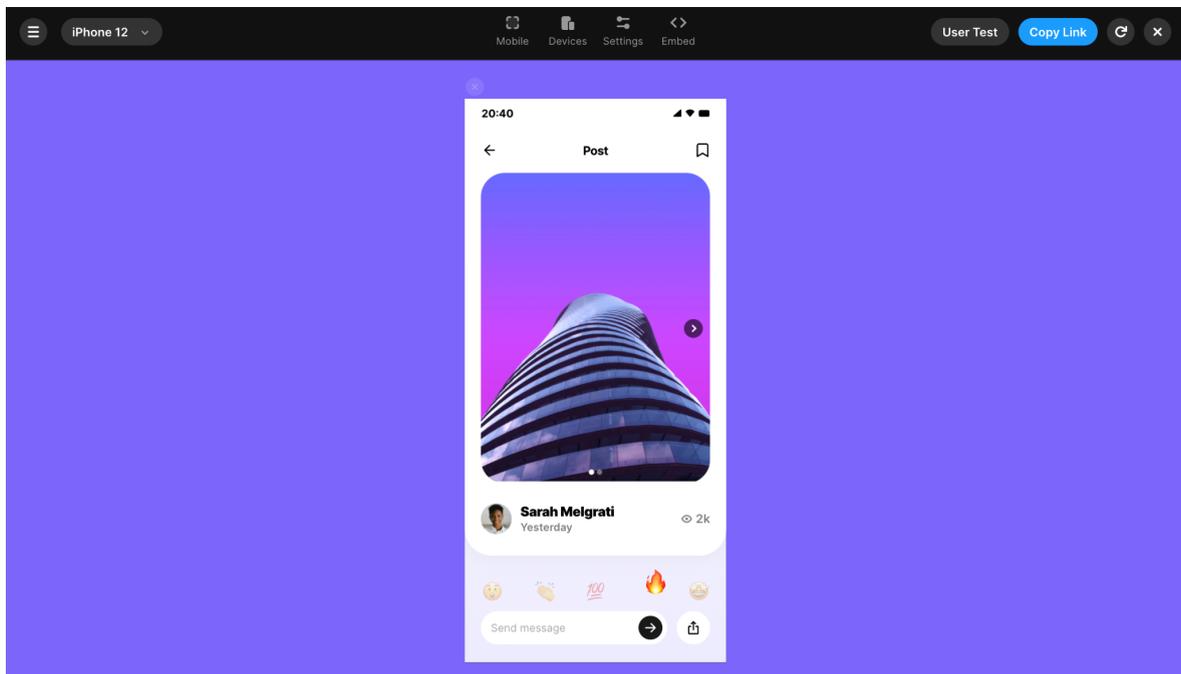
**Figura 56:** Botón para compartir una publicación con un efecto vibrotáctil agregado.

Finalmente el caso de la reacción con el emoticón de fuego fue un tanto más complejo ya que dicha reacción posee también una interacción que hace que al hacer clic sobre el emoticón, el diseño cambie mostrando que esa publicación ya fue reaccionada. Estos cambios de estado se ilustran en las figuras 56 y 57, que muestran cómo se puede visualizar el diseño desde la aplicación de Framer utilizando el modo de “Preview” o vista previa.

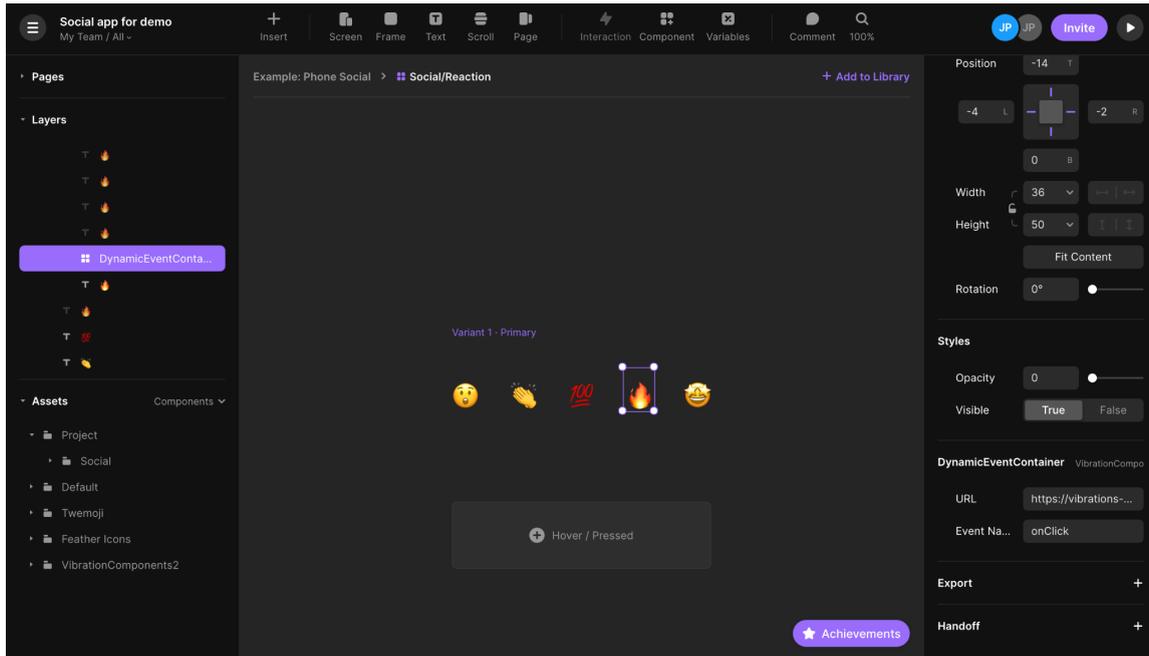
Para poder llevar a cabo esto, fue necesario remover el componente **Stack** de Framer que agrupaba los emoticones y dejarlos sueltos en el diseño, y de esta forma superponer el componente de vibración al emoticón de fuego. El componente agregado sobre el emoticón de fuego puede verse en la figura 58.



**Figura 56:** Emoticones previo a una reacción sobre la publicación.



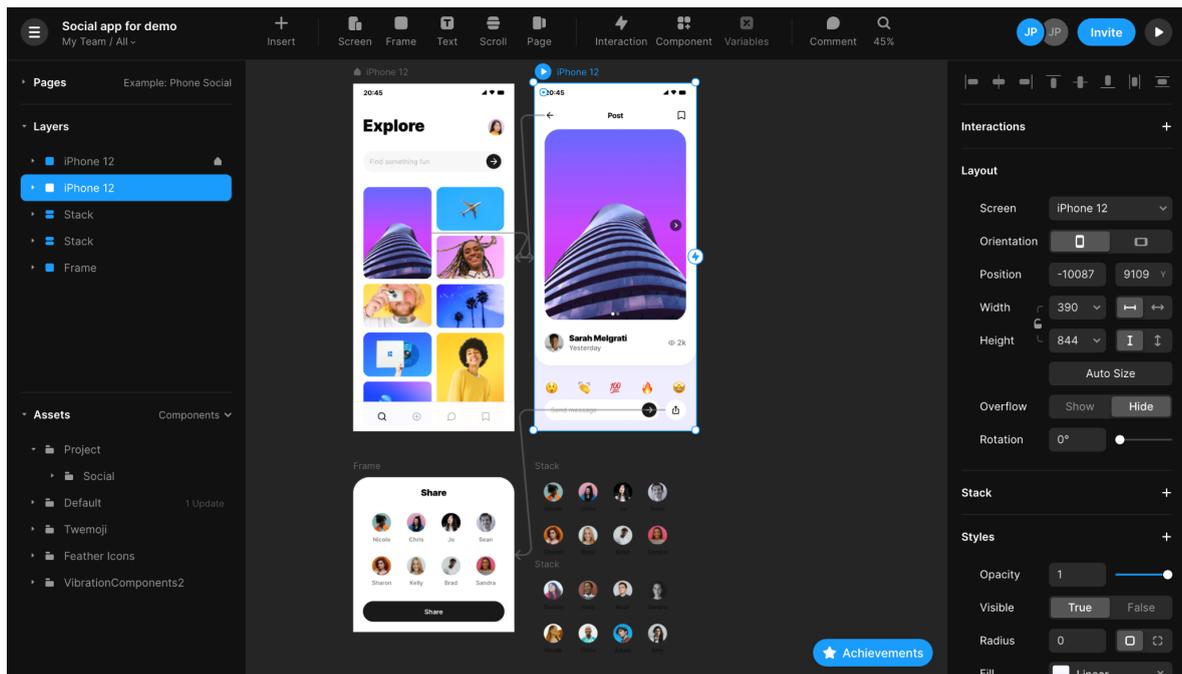
**Figura 57:** Emoticones luego de reaccionar sobre la publicación.



**Figura 58:** Componente para ejecutar vibración agregado sobre emoticón de fuego.

En todos los casos los componentes agregados para reproducir el efecto vibrotáctil fueron configurados con el evento **onClick** y con el enlace correspondiente para reproducir cada vibración, copiado desde la aplicación web.

En cada etapa el diseñador puede hacer una prueba real con un celular. El resultado final del prototipo con todos los efectos incluidos sigue siendo editable desde Framer, como se muestra en la figura 59, que incluye también el flujo de interacciones.

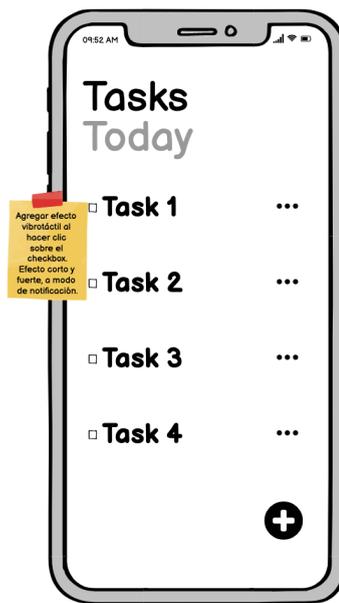


**Figura 59:** Resultado final del prototipo en Framer con todas sus interacciones.

## To do App

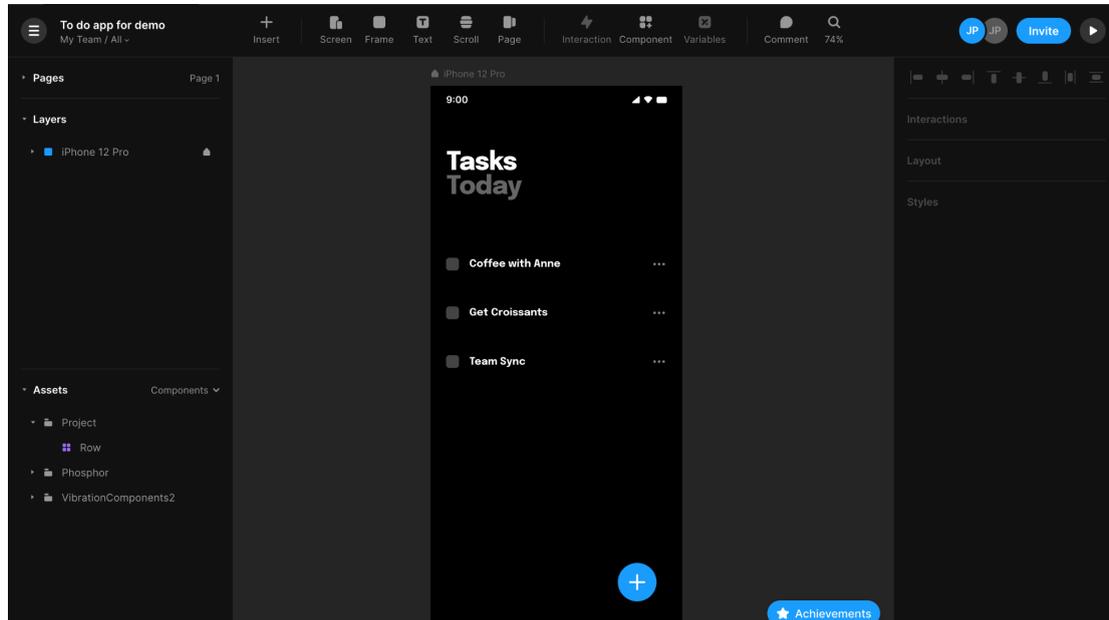
Al igual que con la aplicación Social App, decidimos agregar también efectos vibrotáctiles a la aplicación denominada “To Do App”, presentada en la página de ejemplos de Framer [92]. Este segundo ejemplo presenta una lista de tareas con una casilla para marcarlas como resueltas y luego es posible marcarlas como no resueltas nuevamente. Al hacer clic sobre cada una de las tareas, si estaba como no resuelta será marcada como resuelta y el diseño mostrará la tarea como tachada, y en caso de hacer clic sobre una tarea tachada, será restaurada a su estado original.

Recurriendo nuevamente a un proceso de diseño habitual, es posible que el diseñador genere un prototipo de baja fidelidad del diseño, presentando la herramienta con sus funciones básicas y mostrando la interacción de los usuarios y efectos vibrotáctiles a agregar. Un ejemplo de este prototipo realizado con la herramienta Balsamiq [37] es presentado en la figura 60.



**Figura 60:** Prototipo de baja fidelidad de To Do App.

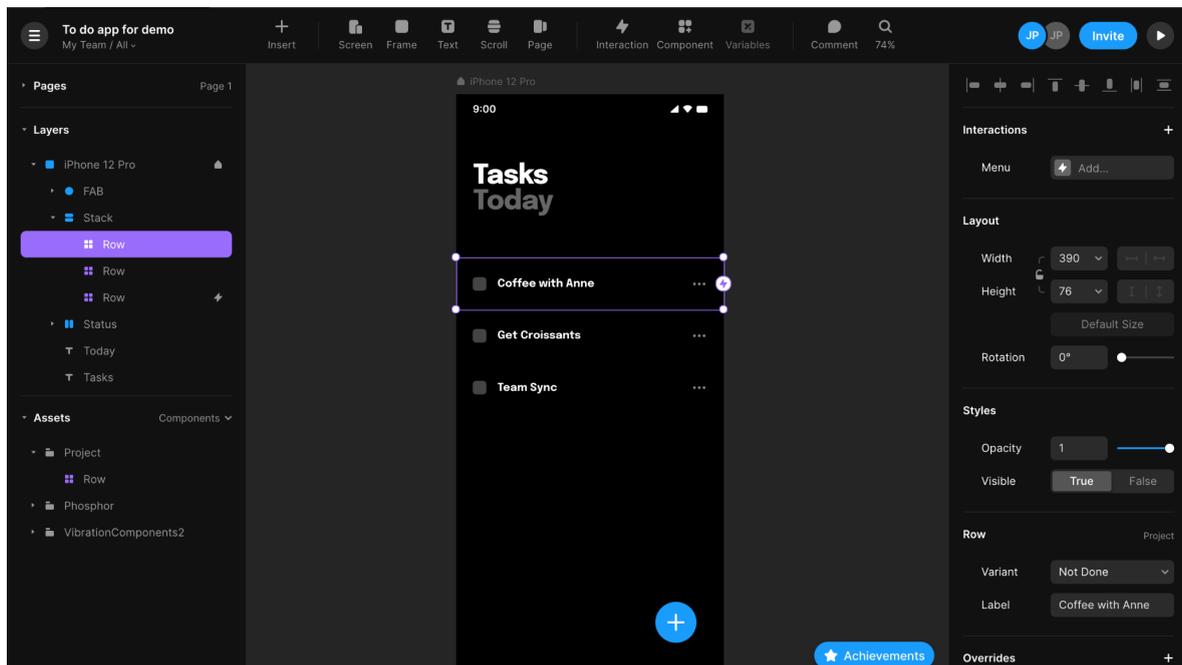
Continuando con el proceso, luego de la validación del prototipo debe crearse otro nuevo prototipo de mayor fidelidad en Framer, previo a comenzar con el desarrollo del mismo. Esta implementación puede verse en la figura 61.



**Figura 61:** Prototipo de alta fidelidad de To Do app implementado en Framer.

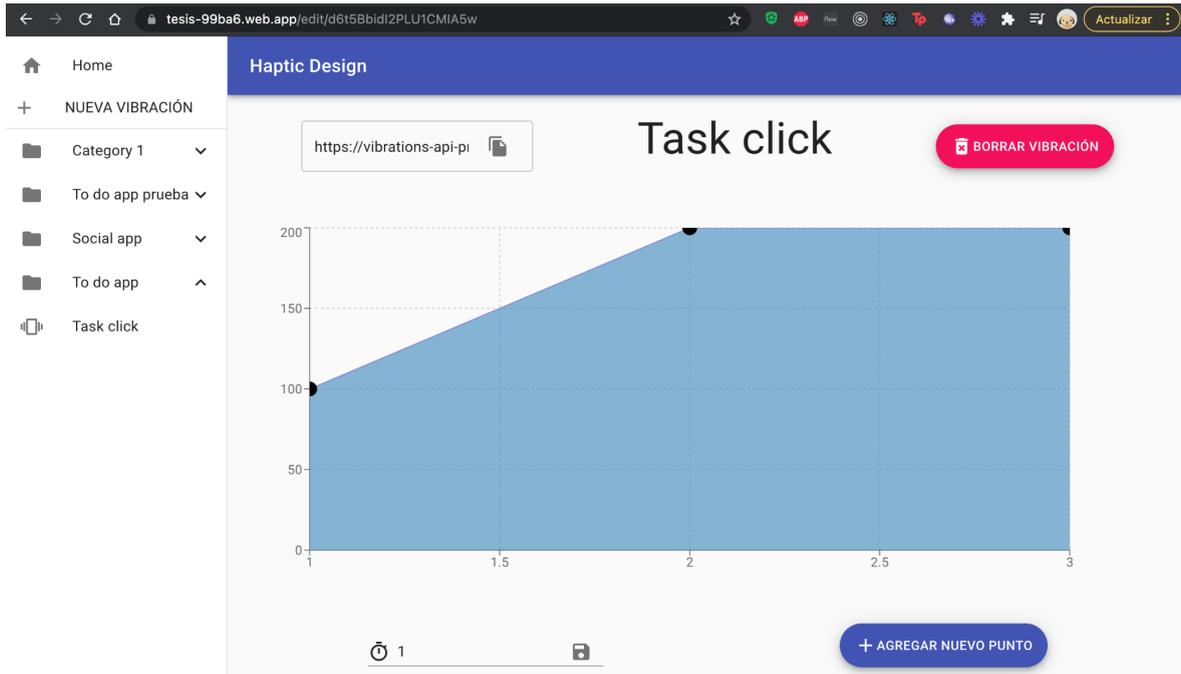
Una vez generado el prototipo que se ajusta al diseño generado en primera instancia, el diseñador buscará agregar los efectos vibrotáctiles. En este ejemplo le agregaremos vibraciones cuando un usuario marca como resuelta o como no resulta una tarea, es decir cuando hace clic sobre cualquier de las tareas que el diseño muestra.

Luego de crear nuestra copia del diseño a partir de la página de Framer e instalar la librería con los componentes para ejecutar las vibraciones, notamos que cada fila con la tarea y el campo checkbox son un componente de Framer, el cual se utiliza 3 veces con un texto distinto, con lo cual solo fue necesario aplicar el componente de vibración en dicho componente y automáticamente cada una de las filas tendría asociada la misma vibración. Los 3 componentes agregados en el diseño pueden verse en la figura 62.



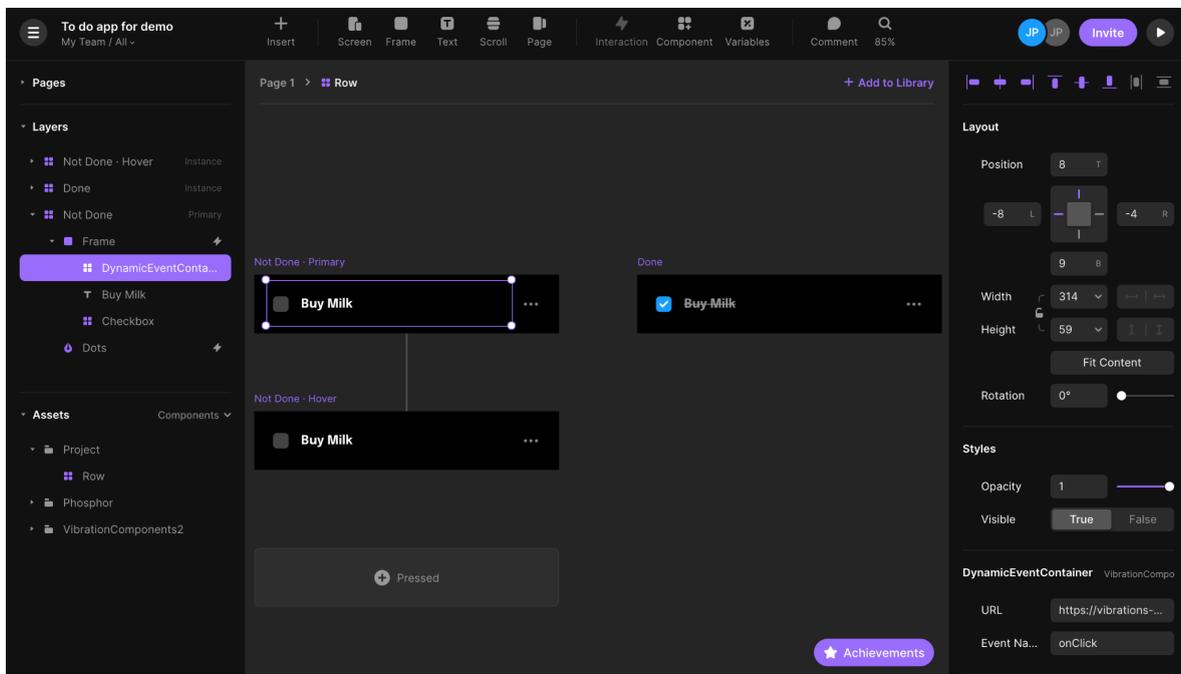
**Figura 62:** Componente “Row” que contiene cada una de las tareas en el diseño To Do App de framer.

Nuevamente generamos una nueva vibración desde el sitio web de diseño de vibraciones [91], para la cual también generamos una categoría llamada **To do app** que contendrá la vibración que reproduciremos al hacer clic sobre las tareas. Esto se muestra en la figura 63.



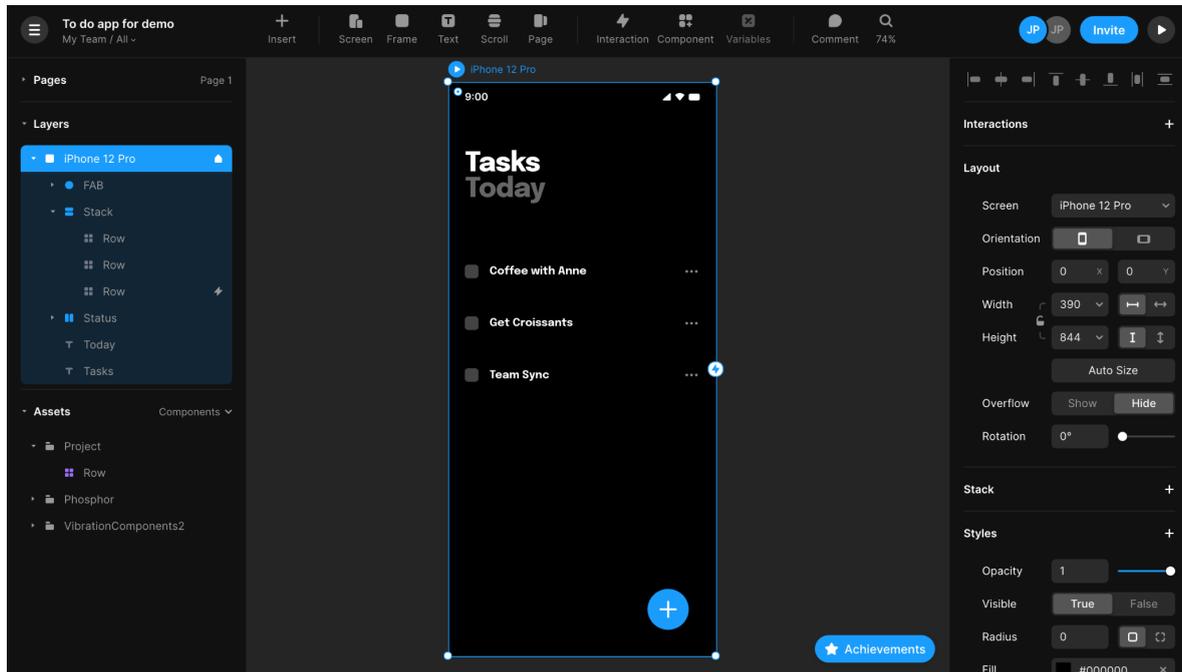
**Figura 63:** Categoría creada para agrupar las vibraciones de la aplicación To do app, junto con la vibración utilizada en el ejemplo.

Al seleccionar la opción de editar el componente **Row**, pudimos sobreponer el componente para ejecutar la vibración fácilmente, configurandolo con el enlace copiado desde la aplicación web para el diseño de la vibración. La figura 64 ilustra al componente Row con el componente para ejecutar la vibración agregada.



**Figura 64:** Componente Row de To Do App junto con el componente para ejecutar el efecto vibrotáctil.

Finalmente, el prototipo final con todas sus interacciones y componentes puede verse en la figura 65. Este prototipo puede seguir siendo editado y probado desde el teléfono como se muestra en los videos.



**Figura 65:** Prototipo final de To Do app en la herramienta Framer.

## Capítulo 4: Conclusiones y posibles trabajos futuros

Se realizó una exhaustiva investigación sobre los progresos actuales en materia de herramientas y tecnologías disponibles para el desarrollo de experiencias hápticas en dispositivos móviles, como también se estudió el proceso de diseño de experiencias de usuario, en busca de generar una herramienta o flujo de trabajo que permita acoplar háptica dentro de dicho proceso, beneficiando a los diseñadores que busquen agregar háptica a sus diseños o generar experiencias más inmersivas con el uso de vibraciones, y también colaborar con los avances de este campo ayudando a la unificación de conceptos y percepciones de las personas frente a las vibraciones. Adicionalmente, en el contexto empresarial, se plantean varios desafíos a la hora de incluir vibraciones en productos desarrollados para clientes, los cuales desencadenan en malas estimaciones de tiempos, desentendimientos, frustraciones y pérdidas de dinero, y este trabajo presenta un posible camino que ayude a tratar estos casos.

Luego de la investigación realizada, se logró establecer un flujo de trabajo que permita a los diseñadores de experiencias de usuario, incluir háptica en sus diseños de interfaces. Gracias a la utilización de la herramienta de prototipado y diseño de interfaces de usuario, Framer, fue posible no solo el acoplamiento de vibraciones a diseños sino también la posibilidad de generar prototipos interactivos que pueden ser probados desde dispositivos móviles, generando vibraciones que permiten dar retroalimentación háptica a los usuarios del sistema a desarrollar.

Se creó una herramienta web que permite la creación de efectos vibrotáctiles mediante la utilización de un gráfico interactivo que ilustra el efecto generado. Esta vibración puede luego ser probada dentro del flujo mencionado anteriormente, permitiendo la posibilidad de retocar o ajustar los parámetros para perfeccionar el efecto que el diseñador trata de generar.

Se espera que este flujo de ejecución pueda ser útil para los diseñadores, y también fácil de integrar en su trabajo diario ya que no requiere el aprendizaje de ningún tipo de herramienta nueva para su diseño de interfaces, y la herramienta desarrollada en este trabajo presenta una interfaz muy simple e intuitiva, por lo que se espera que la curva de aprendizaje no sea alta en absoluto.

Una de las cuestiones a destacar sobre el enfoque tomado en este trabajo es la idea de permitir el desarrollo de interfaces a través de una herramienta web, accesible desde cualquier computadora. Sin embargo, un camino alternativo podría ser el desarrollo de una aplicación nativa de dispositivos móviles, para Android o iOS, que permita la creación y prueba de vibraciones desde el mismo dispositivo sin necesidad de un ordenador adicional. También esta herramienta utiliza un gráfico que permite interpretar una vibración de manera visual, pero podría tomarse un enfoque que genere vibraciones en base a dibujos a mano alzada que realicen los usuarios, o utilizando algoritmos más complejos que estudian la conducta de las personas y generen experiencias acordes y personalizadas para ellos.

Además, este trabajo está orientado a un contexto de aplicaciones móviles pero podría ser fácilmente extendido para abarcar otro tipo de dispositivos, como joysticks de consolas o cualquier otro tipo de dispositivo físico (un enfoque orientado a IoT).

Otra cosa a destacar, es que las herramientas presentadas en este trabajo podrían ser utilizadas para colaborar con cualquiera de los enfoques planteados anteriormente, ya que el servicio backend desarrollado podría ser consumido a través de internet en cualquier dispositivo o desde cualquier interfaz nueva que se desee desarrollar, compartiendo su base de datos de vibraciones ya diseñadas.

## Capítulo 5: Bibliografía

- [1] «Internet de las cosas», *Wikipedia, la enciclopedia libre*. jun. 25, 2021. Accedido: jun. 27, 2021. [En línea]. Disponible en: [https://es.wikipedia.org/w/index.php?title=Internet\\_de\\_las\\_cosas&oldid=136582135](https://es.wikipedia.org/w/index.php?title=Internet_de_las_cosas&oldid=136582135)
- [2] K. E. MacLean, O. S. Schneider, y H. Seifi, «Multisensory haptic interactions: Understanding the sense and designing for it», en *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations-Volume 1*, 2017, pp. 97-142.
- [3] M. Grunwald y M. John, «German pioneers of research into human haptic perception», en *Human Haptic Perception: Basics and Applications*, M. Grunwald, Ed. Basel: Birkhäuser Basel, 2008, pp. 15-39. doi: 10.1007/978-3-7643-7612-3\_2.
- [4] S. J. Lederman y R. L. Klatzky, «Hand movements: A window into haptic object recognition», *Cognit. Psychol.*, vol. 19, n.º 3, pp. 342-368, 1987, doi: [https://doi.org/10.1016/0010-0285\(87\)90008-9](https://doi.org/10.1016/0010-0285(87)90008-9).
- [5] «A. Okamura, J. Dennerlein, and R. Howe. Vibration feedback models for virtual environments. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation, volume 1, pages 674–679. IEEE, 1998. ISBN 0-7803-4300-X. doi:10.1109/ROBOT.1998.677050». Accedido: may 29, 2021. [En línea]. Disponible en: <http://www.springerlink.com/index/10.3758/APP.71.7.1439>.
- [6] P. BACH-Y-RITA, C. C. COLLINS, F. A. SAUNDERS, B. WHITE, y L. SCADDEN, «Vision Substitution by Tactile Image Projection», *Nature*, vol. 221, n.º 5184, pp. 963-964, mar. 1969, doi: 10.1038/221963a0.
- [7] P. Bach-y-Rita, M. E. Tyler, y K. A. Kaczmarek, «Seeing with the Brain», *Int. J. Human-Computer Interact.*, vol. 15, n.º 2, pp. 285-295, abr. 2003, doi: 10.1207/S15327590IJHC1502\_6.
- [8] «D. Prescher, G. Weber, and M. Spindler. A tactile windowing system for blind users. In Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility, 12th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS'10, pages 91–98, New York, New York, USA, 2010. ACM. ISBN 9781605588810. doi:10.1145/1878803.1878821». Accedido: may 29, 2021. [En línea]. Disponible en: <http://www.scopus.com/inward/record.url?eid=2-s2.0-78650615361{&}%20partnerID=40{&}md5=03aab42397cfae7514aa54752490f023http:%20//portal.acm.org/citation.cfm?doid=1878803.1878821>
- [9] J. C. Bliss, M. H. Katcher, C. H. Rogers, y R. P. Shepard, «Optical-to-Tactile Image Conversion for the Blind», *IEEE Trans. Man-Mach. Syst.*, vol. 11, n.º 1, pp. 58-65, 1970, doi: 10.1109/TMMS.1970.299963.
- [10] C. V. Jansson-Boyd, «Touch matters: exploring the relationship between consumption and tactile interaction», *Soc. Semiot.*, vol. 21, n.º 4, pp. 531-546, 2011, doi: 10.1080/10350330.2011.591996.
- [11] «S. Papert. Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, 2nd edition, 1980. ISBN 0465046274». Accedido: may 29, 2021. [En línea]. Disponible en: <http://books.google.com/books?id=HhIEAgUfGHwC&pgis=1>
- [12] E. Gunther, G. Davenport, y S. O'Modhrain, «Cutaneous grooves: composing for the sense of touch», en *Proceedings of the 2002 conference on New interfaces for musical expression*, SGP, may 2002, pp. 1-6.
- [13] A. Chan, K. MacLean, y J. McGrenere, «Designing haptic icons to support collaborative turn-taking», *Int. J. Hum.-Comput. Stud.*, vol. 66, n.º 5, pp. 333-355, 2008, doi: <https://doi.org/10.1016/j.ijhcs.2007.11.002>.
- [14] M. Pielot, N. Henze, y S. Boll, «Supporting map-based wayfinding with tactile cues», en *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, New York, NY, USA, sep. 2009, pp. 1-10. doi: 10.1145/1613858.1613888.

- [15] C. Spence y A. Gallace, «Multisensory design: Reaching out to touch the consumer», *Psychol. Mark.*, vol. 28, n.º 3, pp. 267-308, 2011, doi: 10.1002/mar.20392.
- [16] «Tacto». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://es.wikipedia.org/wiki/Tacto>
- [17] «Háptica». Accedido: abr. 18, 2021. [En línea]. Disponible en: [https://es.wikipedia.org/wiki/H%C3%A1ptica#:~:text=El%20t%C3%A9rmino%20h%C3%A1ptica%20E2%80%8B%20designa,%22%2C%20relativo%20al%20tacto\)](https://es.wikipedia.org/wiki/H%C3%A1ptica#:~:text=El%20t%C3%A9rmino%20h%C3%A1ptica%20E2%80%8B%20designa,%22%2C%20relativo%20al%20tacto)).
- [18] «Cinestesia», *Wikipedia, la enciclopedia libre*. mar. 19, 2021. Accedido: jun. 27, 2021. [En línea]. Disponible en: <https://es.wikipedia.org/w/index.php?title=Cinestesia&oldid=134109221>
- [19] L. A. Jones, *Haptics*. MIT Press Essential Knowledge series, 2018.
- [20] «Estímulos mecánicos». Accedido: may 08, 2021. [En línea]. Disponible en: [http://200.23.188.74/sites/ciencia/volumen2/ciencia3/073/htm/sec\\_8.htm#:~:text=El%20est%C3%ADmulo%20mec%C3%A1nico%20consiste%20en,una%20fuerza%20sobre%20la%20mesa](http://200.23.188.74/sites/ciencia/volumen2/ciencia3/073/htm/sec_8.htm#:~:text=El%20est%C3%ADmulo%20mec%C3%A1nico%20consiste%20en,una%20fuerza%20sobre%20la%20mesa).
- [21] O. S. Schneider y K. E. MacLean, «Improvising design with a haptic instrument», en *2014 IEEE Haptics Symposium (HAPTICS)*, 2014, pp. 327-332.
- [22] «Oliver Schneider - Haptic Jazz». Accedido: jun. 27, 2021. [En línea]. Disponible en: <http://oliverschneider.ca/haptic-jazz/>
- [23] H. Seifi, K. Zhang, y K. E. MacLean, «VibViz: Organizing, visualizing and navigating vibration libraries», en *2015 IEEE World Haptics Conference (WHC)*, 2015, pp. 254-259.
- [24] «VibViz». <https://www.cs.ubc.ca/~seifi/VibViz/main.html> (accedido jun. 27, 2021).
- [25] O. S. Schneider y K. E. MacLean, «Studying design process and example use with Macaron, a web-based vibrotactile effect editor», en *2016 IEEE Haptics Symposium (HAPTICS)*, 2016, pp. 52-58.
- [26] K. Hong, J. Lee, y S. Choi, «Demonstration-Based Vibrotactile Pattern Authoring», en *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, New York, NY, USA, 2013, pp. 219-222. doi: 10.1145/2460625.2460660.
- [27] «EH03-hapticlcon-reprint.pdf». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.cs.ubc.ca/~murphy/cpsc590/EH03-hapticlcon-reprint.pdf>
- [28] H. Seifi, M. Chun, y K. E. Maclean, «Toward affective handles for tuning vibrations», *ACM Trans. Appl. Percept. TAP*, vol. 15, n.º 3, pp. 1-23, 2018.
- [29] «Lofelt Docs». <https://developer.lofelt.com/> (accedido jun. 27, 2021).
- [30] «Nice-Vibrations». Accedido: abr. 25, 2021. [En línea]. Disponible en: <https://nice-vibrations-docs.moremountains.com/>
- [31] «Unity». Accedido: abr. 26, 2021. [En línea]. Disponible en: <https://unity.com/es>
- [32] H. Sharp, Y. Rogers, y J. Preece, *Interaction design: beyond human computer interaction, Fifth edition*, 5th ed. 2019.
- [33] Twitter, LinkedIn, y Facebook, «What is the framework for innovation? Design Council's evolved Double Diamond», *Design Council*, mar. 17, 2015. <https://www.designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond> (accedido jun. 27, 2021).
- [34] «List Of 27 Mobile App Design Prototyping Tools». <https://appinventiv.com/blog/top-mobile-app-prototyping-tools/> (accedido jun. 27, 2021).
- [35] «Sketch». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.sketch.com/>
- [36] «Adobe XD». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.adobe.com/la/products/xd.html>
- [37] «<https://balsamiq.com/>». Accedido: abr. 27, 2021. [En línea]. Disponible en: <https://balsamiq.com/>
- [38] «crea-mockups-balsamiq». Accedido: abr. 27, 2021. [En línea]. Disponible en: <https://www.incubaweb.com/5-herramientas-crear-mockups/crea-mockups-balsamiq/>
- [39] «<https://www.figma.com/>». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.figma.com/>

- [40] «Framer». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.framer.com/why-framer/>
- [41] H. Seifi, M. Chun, C. Gallacher, O. S. Schneider, y K. E. MacLean, «How Do Novice Hapticians Design? A Case Study in Creating Haptic Learning Environments», *IEEE Trans. Haptics*, 2020.
- [42] H. Seifi, *Personalizing Haptics*. 2019. [En línea]. Disponible en: <http://link.springer.com/10.1007/978-3-030-11379-7>
- [43] «Vibration\_API». Accedido: abr. 18, 2021. [En línea]. Disponible en: [https://developer.mozilla.org/en-US/docs/Web/API/Vibration\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Vibration_API)
- [44] «<https://www.figma.com/>». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.figma.com/>
- [45] «Figma plugins development». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.figma.com/plugin-docs/intro/>
- [46] «HTML canvas Tag». [https://www.w3schools.com/tags/tag\\_canvas.asp](https://www.w3schools.com/tags/tag_canvas.asp) (accedido jun. 27, 2021).
- [47] «Figma events». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.figma.com/plugin-docs/api/properties/figma-on/#available-event-types>
- [48] «Adobe XD». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.adobe.com/la/products/xd.html>
- [49] «Adobe XD plugins API». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://adobexdplatform.com/plugin-docs/>
- [50] «Accessing APIs · Adobe XD Plugin Reference». <https://adobexdplatform.com/plugin-docs/reference/core/apis.html> (accedido jun. 27, 2021).
- [51] «Sketch». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.sketch.com/>
- [52] «Plugins», *Sketch Developers*. <https://developer.sketch.com/plugins/> (accedido jun. 27, 2021).
- [53] «CocoaScript», *Sketch Developers*. <https://developer.sketch.com/plugins/cocoascript> (accedido jun. 27, 2021).
- [54] «SPKM». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://github.com/skpm/skpm>
- [55] «What are Frameworks?» <https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/BPFrameworks/Concepts/WhatAreFrameworks.html> (accedido jun. 27, 2021).
- [56] «Apple audio services». Accedido: abr. 18, 2021. [En línea]. Disponible en: <http://iphonedevwiki.net/index.php/AudioServices>
- [57] «Audio Toolbox | Apple Developer Documentation». <https://developer.apple.com/documentation/audiotoolbox> (accedido jun. 27, 2021).
- [58] «Framer». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.framer.com/why-framer/>
- [59] «React – Una biblioteca de JavaScript para construir interfaces de usuario». <https://es.reactjs.org/> (accedido jun. 27, 2021).
- [60] «Framer packages». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.framer.com/support/using-framer/packages/>
- [61] «Framer Desktop». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.framer.com/support/using-framer/web-framer-desktop/>
- [62] «Enabling Code In Framer | Framer». <https://www.framer.com/support/using-framer/using-code/> (accedido jun. 27, 2021).
- [63] «Framer Overrides». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.framer.com/support/using-framer/code-overrides/>
- [64] «override-property-controls». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://spectrum.chat/framer/learn/override-property-controls~af9dc47f-20d4-47b9-ae0e-bfbd6de6122f>
- [65] «the-haptic-stack». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.immersion.com/the-haptic-stack/>

- [66] «the-haptic-stack-design-layer». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.immersion.com/the-haptic-stack-design-layer/>
- [67] «Envelope (music)», *Wikipedia*. abr. 25, 2021. Accedido: jun. 27, 2021. [En línea]. Disponible en: [https://en.wikipedia.org/w/index.php?title=Envelope\\_\(music\)&oldid=1019863105](https://en.wikipedia.org/w/index.php?title=Envelope_(music)&oldid=1019863105)
- [68] «the-haptic-stack-software-layer». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.immersion.com/the-haptic-stack-software-layer/>
- [69] «Vibrator | Desarrolladores de Android», *Android Developers*. <https://developer.android.com/reference/android/os/Vibrator?hl=es-419> (accedido jun. 27, 2021).
- [70] «OpenXR». Accedido: abr. 25, 2021. [En línea]. Disponible en: <https://www.khronos.org/registry/OpenXR/specs/1.0/html/xrspec.html>
- [71] «the-haptic-stack-hardware-layer». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.immersion.com/the-haptic-stack-hardware-layer/>
- [72] «About W3C». <https://www.w3.org/Consortium/> (accedido jun. 27, 2021).
- [73] «W3C vibration API». Accedido: may 02, 2021. [En línea]. Disponible en: <https://www.w3.org/TR/vibration/>
- [74] J. H. McClellan, R. W. Schafer, y M. A. Yoder, *Dsp first*. Pearson Education, 2017.
- [75] «PWM wikipedia». Accedido: may 06, 2021. [En línea]. Disponible en: [https://en.wikipedia.org/wiki/Pulse-width\\_modulation#Duty\\_cycle](https://en.wikipedia.org/wiki/Pulse-width_modulation#Duty_cycle)
- [76] «Express - Node.js web application framework». <https://expressjs.com/> (accedido jun. 27, 2021).
- [77] «Firebase». Accedido: may 02, 2021. [En línea]. Disponible en: <https://firebase.google.com/>
- [78] «Firebase pricing». Accedido: may 02, 2021. [En línea]. Disponible en: <https://firebase.google.com/pricing?hl=es-419>
- [79] «Heroku». Accedido: may 02, 2021. [En línea]. Disponible en: <https://www.heroku.com/>
- [80] «ReactJS». Accedido: may 06, 2021. [En línea]. Disponible en: <https://reactjs.org/>
- [81] «Framer Pricing». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.framer.com/pricing/>
- [82] «New desktop apps for Windows and Mac | Framer». <https://www.framer.com/support/using-framer/windows-mac/> (accedido jun. 27, 2021).
- [83] «folder-backed-projects». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.framer.com/support/using-framer/folder-backed-projects/>
- [84] «framer-cli». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.npmjs.com/package/framer-cli>
- [85] D. Ledo, S. Houben, J. Vermeulen, N. Marquardt, L. Oehlberg, y S. Greenberg, «Evaluation Strategies for HCI Toolkit Research», en *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: Association for Computing Machinery, 2018, pp. 1-17. Accedido: jun. 30, 2021. [En línea]. Disponible en: <https://doi.org/10.1145/3173574.3173610>
- [86] M. Wiberg y E. Stolterman, «What makes a prototype novel? a knowledge contribution concern for interaction design research», en *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, New York, NY, USA, oct. 2014, pp. 531-540. doi: 10.1145/2639189.2639487.
- [87] «Framer examples». Accedido: ago. 01, 2021. [En línea]. Disponible en: <https://www.framer.com/examples/>
- [88] «Historias de usuario», *Wikipedia, la enciclopedia libre*. oct. 02, 2020. Accedido: jun. 27, 2021. [En línea]. Disponible en: [https://es.wikipedia.org/w/index.php?title=Historias\\_de\\_usuario&oldid=129725337](https://es.wikipedia.org/w/index.php?title=Historias_de_usuario&oldid=129725337)
- [89] «user-stories». Accedido: abr. 18, 2021. [En línea]. Disponible en: <https://www.atlassian.com/es/agile/project-management/user-stories>
- [90] «Framer Social App». Accedido: ago. 01, 2021. [En línea]. Disponible en: <https://www.framer.com/examples/social-app/>
- [91] «Aplicación de diseño de vibraciones». Accedido: ago. 01, 2021. [En línea]. Disponible

en: <https://tesis-99ba6.web.app/>  
[92] «Framer To Do App». Accedido: ago. 01, 2021. [En línea]. Disponible en:  
<https://www.framer.com/examples/to-do-app/>

# Capítulo 6: Anexos

## Videos de demostración

Para ilustrar el proceso completo de agregado de vibraciones a un diseño utilizando la herramienta Framer, presentamos unos videos que fueron grabados utilizando como ejemplo la aplicación To Do App, presentada en la segunda parte de la sección [3.6 Demostración](#).

El primer video presenta el flujo completo de instalación de los componentes creados en el diseño, junto con la creación del efecto vibrotáctil en la aplicación web y el agregado de estos componentes al diseño: <https://drive.google.com/file/d/1ri8syd8XMsTKYtLUqIYViiCHZe1amAQG/view?usp=sharing>

Presentamos también un segundo video donde se pueden ver las acciones requeridas para poder visualizar el diseño creado y probar los efectos vibrotáctiles en un dispositivo móvil. En el video también se puede ver cómo vibra el dispositivo y como cambia el efecto vibrotáctil al hacer modificaciones desde la aplicación web y refrescar los cambios: [https://drive.google.com/file/d/16ibmfpuywvQfFS\\_iEQo1h8vvHDTL4A84/view?usp=sharing](https://drive.google.com/file/d/16ibmfpuywvQfFS_iEQo1h8vvHDTL4A84/view?usp=sharing)

## Repositorios de código creados

Para realizar la programación de las herramientas presentadas en este trabajo (aplicación frontend y aplicación backend) se utilizó la herramienta de versionado de código Git y se utilizó Github como plataforma para albergar el código fuente.

A continuación se presentan los enlaces a los repositorios

- Aplicación backend: <https://github.com/JulianPasquale/vibrations-api>
- Aplicación frontend: <https://github.com/JulianPasquale/haptic-design-UI>

Además se dejarán disponibles las aplicaciones mencionadas publicadas en la web, utilizando los proveedores de hosting cloud presentados en [3.5 Arquitectura de la solución](#).

- Aplicación backend: <https://vibrations-api-production.herokuapp.com>
- Aplicación frontend: <https://tesis-99ba6.web.app>

En caso de querer ejecutar las aplicaciones localmente, deberá ponerse en contacto para solicitar credenciales de acceso a Firestore y las variables de entorno necesarias.