

Clase de Primero de Aplicaciones

Aula 408

Escuela CEV - Madrid

Vision Test

La aplicación para medir tus dioptrías desde el móvil.



Logo de la aplicación

Resumen

Se va a desarrollar una **aplicación** en la que se realizará un examen de la vista para determinar las dioptrías mediante las tres pruebas más utilizadas en oftalmología: El test de Snellen, el test de Landolt y el test de daltonismo.

Puntos importantes

Claves de mi aplicación:

- Medición de la distancia
- Reconocimiento facial
- Botones con opciones de respuesta
- Test de Snellen
- Test de Landolt
- Test de daltonismo
- Resultados

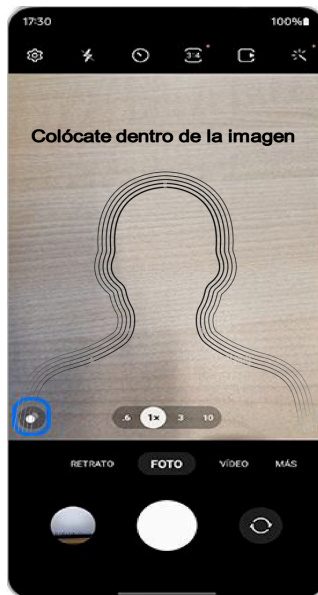
Especificaciones

Medición de la distancia

Se realizará mediante la aplicación de la cámara del dispositivo móvil.

Se colocará una silueta delimitada por líneas discontinuas ya medida para que el usuario coloque su busto dentro de los márgenes.

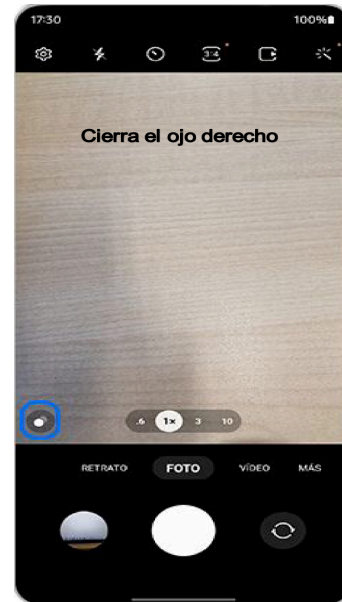
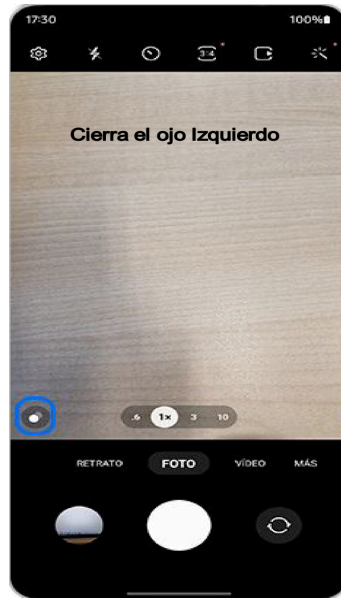
La distancia a la que se colocan los paneles de letras, que se llaman optotipos, es de 6 metros en una consulta. Por lo que se tendría que buscar una medida proporcional para que el usuario no tenga que alejar tanto el móvil y pueda manejar la aplicación con sus manos sin tener que moverse de un sitio a otro. Por lo tanto las letras estarán a un tamaño proporcional.



Reconocimiento facial

Una vez esté colocado el usuario, la aplicación pasará al siguiente paso en el que se le solicitará que se tape el ojo derecho y esta lo reconocerá mediante la implantación de inteligencia artificial, y lo mismo para el ojo izquierdo.

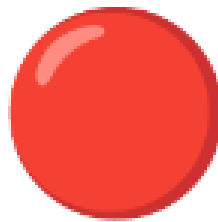
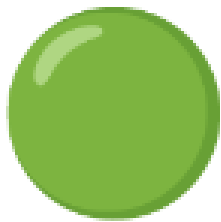
Esto solo será necesario para el test Snellen.



Botones con opciones de respuesta

Se introducirán botones de “sí” o “no” para que el usuario indique si ve la letra en cuestión del test de Snellen y el test de Landolt.

Se introducirán botones con posibles respuestas a modo test (respuesta A, respuesta B y respuesta C) para el test de daltonismo.



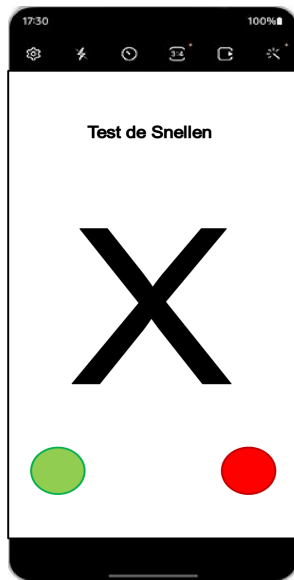
Test de Snellen

Ya tenemos al usuario en posición y con el ojo en cuestión cerrado, por lo que se empieza a realizar el primer test, el test Snellen, que consistirá en ir sacando imágenes de letras en diferentes tamaños, las cuales el usuario responderá, mediante unos botones ya incorporados, si los visualiza o no alcanza a verlos.

Antes de empezar el test la pantalla se queda en blanco para que el ojo haga una especie de reseteo.

Las letras tendrán un sistema aleatorio incorporado a la hora de visualizarse para que el usuario no pueda memorizar las respuestas del test.

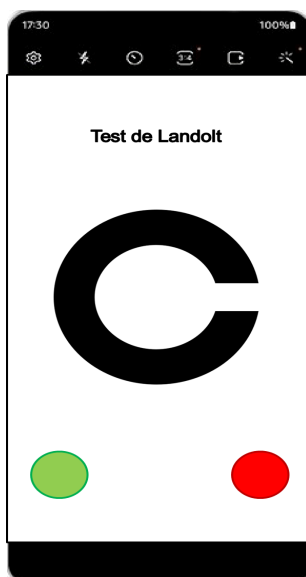
Se realizará la misma prueba con el otro ojo.



Test de Landolt

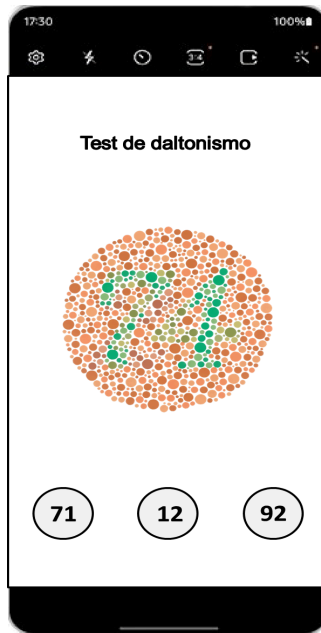
La siguiente prueba consiste en identificar los anillos de Landolt, que son unos caracteres circulares con forma de “o” pero con un trazado no continuo, sino con una discontinuidad cuyo hueco hay que identificar.

El usuario responderá, mediante unos botones ya incorporados, si los visualiza o no alcanza a verlos.



Test de daltonismo

El último test consiste en el visionado de unas imágenes llamadas “placas de color Ishihara”, las cuales el usuario responderá a las diferentes opciones que le ofrecemos del número que se encuentra en la imagen en cuestión.



Resultados

Realizados todos los test solo quedaría la entrega de resultados, los cuales siguen los criterios de oftalmología, pero se le indicará al usuario mediante un mensaje debajo de los resultados que estos son solamente orientativos y que se dirija a una clínica para comprobar si los datos son ciertos.

Además te recomienda realizar todas estas pruebas una vez al año para comprobar si han aumentado o se han mantenido las dioptrías y así asegurarnos de que no va a desinstalar la app .



Modelos de desarrollo

Cascada

El proyecto se crea a partir de distintas fases, las cuales se suceden una detrás de otra y que se ejecutan una sola vez.

Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente.

Este modelo se compone de 5 fases:

1. Análisis.
2. Diseño.
3. Implementación.
4. Verificación.
5. Mantenimiento.



1. Análisis:

En esta fase se estudia la viabilidad del proyecto que se quiere llevar a cabo y para ello se evalúan los costes que conlleva la realización del mismo y a su vez si es rentable y si es factible.

En este proyecto habrá que establecer una serie de requisitos tanto de sistema como de

software y un análisis de estos requisitos que sirva a modo de estrategia para cuando haya problemas poder solventarlos mediante pequeñas tareas.

2. Diseño:

Se podría traducir esta fase como la arquitectura de software, en la que se realiza una planificación detallada de cómo van a ser las interfaces y los entornos de trabajo entre otras muchas cosas.

En esta fase obtendremos un borrador preliminar para llevar a cabo el diseño del software y realizar pruebas en los diferentes componentes.

3. Implementación:

En la fase de implementación se ejecutará toda esa arquitectura de software que se especificó en la fase de diseño, además, se incluirá la programación de dicho software y se buscarán los posibles errores que pueda presentar.

4. Verificación:

También llamada fase de prueba, es la fase donde se envía una versión beta a una serie de usuarios finales, escogidos previamente, para estudiar si obtiene la aceptación que se buscaba y en el caso de que así sea preparar el proyecto para su lanzamiento.

5. Mantenimiento:

También llamada fase de servicio, en la que se lanza el proyecto pero además se supervisa que funcione correctamente y en el caso de haber desperfectos poder mejorarlos introduciendo nuevas versiones.

Espiral

El proceso sigue una serie de etapas, empezando por la de conceptualización y siguiendo por la de desarrollo para ver más tarde las posibles mejoras que podemos incorporar en la etapa de mejoras y acabar con la de mantenimiento.

En cada una de las etapas hay una serie de fases:

1. Planificación.
2. Análisis de riesgos.
3. Desarrollo.
4. Evaluación

Se añade además una fase de enlace entre las distintas etapas, para que la transición entre ellas se haga más fácil.

1. Planificación:

Se estima el coste, el calendario y los recursos necesarios para el proyecto.

2. Análisis del riesgo:

Se realiza una investigación en búsqueda de los posibles riesgos, a la vez que se planifica una estrategia para poder solventarlos.

3. Ingeniería:

Incluye la codificación, pruebas y el despliegue del software.

4. Evaluación:

El cliente evalúa el software.

También se indaga sobre los riesgos que pueda tener dicho software como retrasos en los plazos o si va a haber algún sobrecoste.

Cascada vs Espiral

El modelo de cascada es muy útil en proyectos pequeños con unos requisitos muy claros, mientras que el modelo en espiral es una mejor opción en cuanto al desarrollo de proyectos más grandes y

complicados que necesitan continuamente analizar los riesgos.

Si se trata de un proyecto con las bases claras y con los requisitos ya establecidos y que no conlleva muchos cambios sería más prudente escoger un modelo en cascada. En cambio, si se trata de un proyecto más flexible en cuanto a los requisitos y en el que poder realizar cambios a medida que van apareciendo posibles mejoras sería conveniente utilizar el método en espiral.

Los costos en el modelo de cascada son más económicos, mientras que en el modelo de espiral son más altos.

En cuanto a la ejecución del proyecto, el modelo en espiral permite crear prototipos, con la finalidad de ir probando el producto final antes de que este sea lanzado. En cambio en el modelo en cascada no puedes saber cómo quedará el producto final hasta que el proyecto se acabe.

Secuencial

El proyecto se divide en varias partes y luego cada parte se desarrolla una por una.

Cada una de estas partes, en las que se divide el proyecto, proporciona funcionalidad que cumple con los requisitos generales del proyecto. Esto permite comprobar los resultados más rápido si lo comparamos con el modelo en cascada.

El producto final se construye progresivamente.

Iterativo

Primero se crea un esquema sobre la base para poder comprender cómo va a comenzar el proyecto y cómo se va a hacer, más tarde se realizan las mejoras en varios pasos.

Es un proceso en que se experimenta continuamente, se realizan pruebas y se evalúa el producto, de esta forma consigue eliminar los problemas progresivamente y se amolda para los usuarios, que van a ser quienes lo utilicen.

Con este modelo conseguimos mejorar el diseño a lo largo del tiempo.



Secuencial vs Iterativo

Estos dos modelos de desarrollo se centran en la entrega rápida y continua.

Se pueden utilizar juntos o pueden combinarse creando lo que se llama “modelo ágil” para poder crear estrategias de crecimiento más efectivas.

La elección de cada método depende de lo complejo que sea el proyecto, así como de los requisitos y la experiencia del equipo de desarrollo.

Modelo de desarrollo Vision Test

Ahora que ya hemos comprendido los diferentes tipos de modelos de desarrollo que se pueden aplicar a nuestra aplicación podemos tomar una conclusión.

La mejor opción para el desarrollo de nuestra aplicación es un modelo en cascada y secuencial, ya que las bases están muy claras y no se trata de un proyecto grande. Además, no podemos olvidar que el modelo en cascada es más económico.

Esperaremos para saber cómo quedará el producto final y si hay que realizar algún pequeño cambio lo haremos, pero serán cambios a nivel de estética o adaptación para que se adecue al cliente.



Formas de distribución

Para poder hablar de distribución hay que primero establecer qué canal se va a utilizar.

Hay 2 tipos de canales de distribución:

Directo

El productor o la empresa productora son los que se encargan de que el producto llegue al consumidor final sin necesidad de utilizar intermediarios.

Indirecto

El productor o la empresa productora requieren de uno o varios intermediarios para poder llevar el producto al consumidor final.

Hay 2 tipos de canales indirectos:

- **Corto:** el número de intermediarios es muy reducido o solamente un único intermediario.
- **Largo:** se requiere a varios intermediarios.

En cuanto a los canales de distribución podemos hablar de **tiendas físicas, sitios web, call centers, aplicaciones móviles, redes de distribuidores, ferias comerciales y eventos.**

Después de tomar la decisión de qué canal de distribución utilizar, se crea una estrategia de distribución.

Hay 3 tipos de estrategias de distribución:

Intensiva

El productor o la empresa productora quieren llevar su producto al mayor número de puntos de venta posible. Esto requiere de tantos intermediarios como puntos de venta se escojan.

Exclusiva

El productor o la empresa productora solo requieren a un intermediario, pero uno en cada zona. La zona podría tratarse de regiones o de ciudades.

Este modelo de distribución es para productos muy caros y exclusivos.

Selectiva

El productor o la empresa productora requieren de un número reducido de intermediarios. En esta estrategia el papel y reputación del intermediario es fundamental de cara a la venta del producto.

En cuanto a los factores que afectan a la estrategia de distribución podemos hablar los ***hábitos de consumo***, las propias ***características del producto***, como está organizado el ***mercado*** o la ***estructura de negocio***.

Forma de distribución Vision Test

Para poder llegar al público más amplio posible necesito distribuirlo en un mercado como el de Google Play.

Google Play es el mercado principal en cuanto a lo que se trata de aplicaciones para Android.

En el caso de ver que la aplicación me está reportando un dinero no estimado, podría empezar a pensar otras formas de distribución más entramadas como por ejemplo sacar una versión para Apple o incluso recurrir a medios de comunicación como la televisión, cartelería o merchandising.

Otra forma para poder dar a conocer mi aplicación sería apuntarse a eventos específicos donde la gente venga a conocer sobre aplicaciones móvil.



Planificación

Para poder hablar de planificación hay que primero establecer las limitaciones y objetivos.

Acabaremos con una presentación.

Limitaciones

- **Tiempo:**

- **Nº de horas disponibles:** Se dispone de un total de 800 horas para el desarrollo, pruebas y ajustes finales.
- **Fechas de entrega:** La fecha límite para la prueba beta de la aplicación es el 15 de marzo de 2024, y la versión final se entregará el 30 de abril de 2024.
- **Horarios de disponibilidad:** El equipo estará disponible de lunes a viernes, de 9:00 a.m. a 6:00 p.m., con flexibilidad para programar reuniones adicionales según sea necesario.
- **Vacaciones:** Se programarán las vacaciones del equipo.

- **Recursos:**

- **Hardware:** Equipos y procesadores serán utilizados para el desarrollo y pruebas.
- **Espacios:** Se designarán espacios de trabajo colaborativo y se programarán salas de reuniones para sesiones de planificación y revisión.
- **Software:** Se utilizarán herramientas como Android Studio para la comunicación interna.

- **Comunicación:** Se llevarán a cabo reuniones semanales los viernes y se usará un correo interno para comunicaciones diarias y ágiles.

- **Personal:**

- **Desarrolladores:** Se designarán tres desarrolladores con experiencia en el desarrollo de aplicaciones móviles.
- **Asesores y Expertos:** Contararemos con un experto en accesibilidad visual para garantizar la calidad de las pruebas de graduación de la vista.
- **Gestión de ausencias:** Se establecerá un sistema de rotación de funciones para cubrir ausencias.

Objetivos

- **Versiones:**

- **Beta:**
 - **Tema:** Desarrollo básico de la aplicación con capacidad para realizar pruebas de visión iniciales.
 - **Épica:** Creación de la estructura principal y la interfaz de usuario.
 - **Historia de Usuario:** Poder realizar una prueba de agudeza visual básica.
- **Entrega Final:**
 - **Tema:** Mejora de funcionalidades y corrección de errores.
 - **Épica:** Implementación de pruebas de graduación de la vista más avanzadas.
 - **Historia de Usuario:** Realizar exámenes exhaustivos de la vista y recibir análisis detallados de mis resultados, así como sugerencias para mejorar mi salud visual.

- **Tareas:**

- ***Pruebas Visuales:***

- Desarrollar algoritmos para pruebas de agudeza visual.
 - Implementar pruebas de graduación de la vista detalladas.

- ***Interfaz:***

- Diseñar una interfaz intuitiva y amigable.
 - Presentar de forma clara y comprensible los resultados de las pruebas integradas.

- ***Dependencias:***

- Identificar la importancia de finalizar el módulo de pruebas visuales antes de avanzar con la interfaz de usuario.
 - Asegurar la integración de los algoritmos de prueba antes de comenzar las pruebas de usuario.

- **Priorización (modelo MoSCoW):**

- ***Must Have:***

- Implementación de pruebas de agudeza visual.
 - Diseño de interfaz de usuario.
 - Resultados claros de las pruebas.

- ***Should Have:***

- Desarrollo de pruebas de graduación de la vista detalladas.

- Análisis detallado de resultados.
- ***Could Have:***
 - Características complementarias recomendadas por profesionales de la salud visual.
- ***Won't Have (por ahora):***
 - Aspectos que no generan valor de forma inmediata o que podrían ser postergados para actualizaciones futuras.
- **Estimación de tareas:**
 - Empleo de técnicas como Planning Poker para estimar el esfuerzo necesario en cada tarea.
 - Definición de tallas o puntos de historia para evaluar la complejidad de las tareas.
 - Votaciones para alcanzar consensos sobre la prioridad y la estimación del esfuerzo de las tareas.

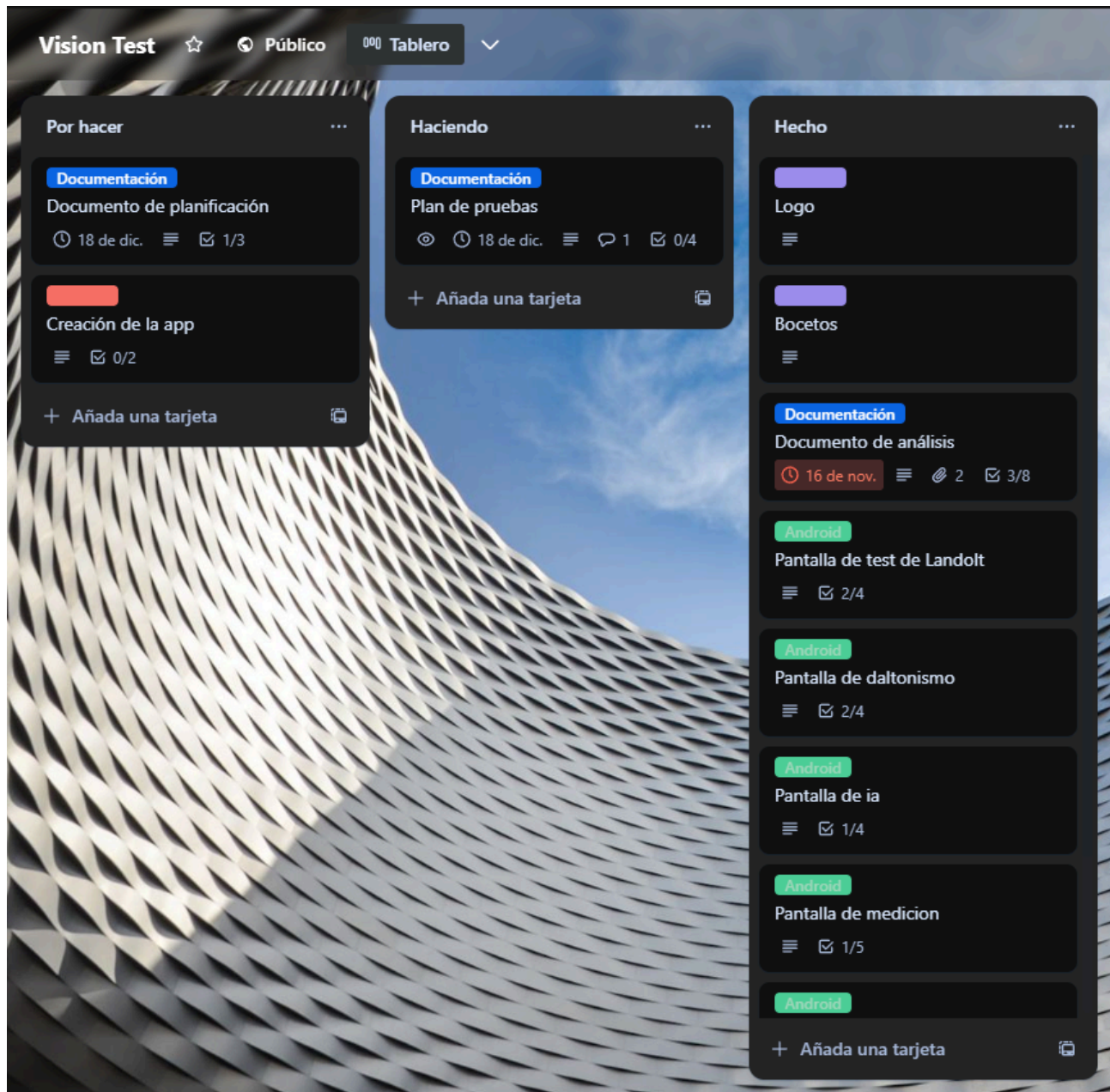
Presentación

- **Diagrama de Gantt:**

<https://app.asana.com/0/1206180073270392/1206180149714413>

- **Trello:**

<https://trello.com/b/uuEKG7Lj/vision-test>



Webgrafía

Vision Test

- https://www.quimica.es/enciclopedia/Test_de_Landolt.html
- <https://www.visioncenter.com.pe/blogs/lentes-ofthalmicos/dioptrias-tipos-graduacion>
- <https://www.saludsavia.com/contenidos-salud/otros-contenidos/test-de-snellen>
- <https://www.aao.org/salud-ocular/enfermedades/como-se-hacen-las-pruebas-para-detectar-daltonismo>
- <https://ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/#:~:text=Contin%C3%BAa-.%C2%BFQu%C3%A9%20es%20el%20modelo%20en%20cascada%3F,ejecuta%20tan%20solo%20una%20vez.>

Modelo de desarrollo

- <https://sentr.io/blog/diferencias-entre-desarrollo-iterativo-e-incremental/>
- <https://glosario.es/istqb/nivel-basico/modelo-de-desarrollo-secuencial/#:~:text=Un%20tipo%20de%20modelo%20de,sucesivas%20sin%20superposici%C3%B3n%20entre%20ellas.>
- <https://ravisr16.wordpress.com/2014/06/22/agile-development-environment/>
- <https://proyectosagiles.org/desarrollo-iterativo-incremental/>
- <https://www.prosci.com/blog/aligning-the-adkar-model-with-sequential-iterative-and-hybrid-change>
- <https://alaimolabs.com/es/self-learning/scrum/gestion-de-proyectos-que-es-el-modelo-secuencial-waterfall-y-como-functiona-este-metodo-en-cascada>
- <https://modelosprocesosdesoftware.blogspot.com/p/modelo-en-cascada-o-lineal-secuencial.html>

Formas de distribución

- <https://www.zendesk.com.mx/blog/estrategias-de-canales-de-distribucion-cuales-son-y-como-elegir/#:~:text=Hay%20tres%20tipos%20de%20estrategia,comercializar%20un%20producto%20en%20particular.>

Planificación

- <https://redwerk.es/blog/cuanto-tiempo-se-tarda-en-crear-una-aplicacion/>
- <https://www.yeeply.com/blog/profesionales-en-el-desarrollo-de-aplicaciones-movil/>
- <https://distillery.com/es/blog/5-pasos-para-planificar-con-exito-el-desarrollo-de-aplicaciones/>
- <https://www.itdo.com/blog/moscow-que-es-y-como-priorizar-en-el-desarrollo-de-tu-aplicacion/>