# *Product Manager*

## **Clase 11 - Software Development**

Objetivo de la clase:

* Comprender cuál es la relación y el trabajo colaborativo con el equipo de desarrollo.
* Conocer el roadmap de trabajo.
* Comprender las diferencias entre las plataformas de un producto digital.

## **Brief: Desarrollo**

Para lanzar un producto debemos contemplar cuestiones de desarrollo a la hora de finalizar nuestro Brief. En esta clase veremos los criterios de desarrollo más importantes a incluir en este documento fundamental. A la vez, será de gran ayuda para la segunda pre entrega del proyecto final. Es importante optimizar el brief con todo lo necesario para que el squad de programadores (y otros miembros clave como UXers, por ejemplo) esté al tanto de los objetivos principales del producto.

Esta clase no tiene por objetivo que aprendas a programar, sino que seas capaz de identificar las herramientas fundamentales a tener en cuenta para el desarrollo, qué necesitas de ellas y en qué medida.

**CAMINO AL DESARROLLO: TRABAJO DEL PRODUCT MANAGER CON DESARROLLADORES**

1. **Brief inicial:** Requerimientos del cliente. Ideas de funcionalidades para personas usuarias.
2. **Pantallas y objetivos:** Lista de pantallas y sus componentes. Evaluar con el cliente. Objetivos de negocio a cumplir.
3. **Presupuesto y tiempo:** El cálculo es en base a la experiencia de desarrollar funcionalidades claves.
4. **Pre-producción:** Wireframes, API’s y Bases de datos.
5. **Producción y programación** Diseño final (Mobile y/o Desktop). Desarrollo Front y Backend.
6. **Testeo del prototipo:** Testeo con usuarios, mejoras, lanzamientos y aprendizaje.

## **Pre-producción**

Llamamos así al proceso previo a programar el código del producto que queremos lanzar. Antes de comenzar con esa tarea de desarrollo, debemos asegurarnos de aspectos como diseño de pantallas y arquitectura de software. Se repite cada vez que se desea generar una nueva funcionalidad para el producto. Se trabaja con desarrolladores, UXers y otras personas del squad, sobre los siguientes aspectos: Wireframes en baja y/o alta calidad, Arquitectura de API’s y Bases de datos. Haz [clic aquí](https://view.genial.ly/62cd8df3f683640011e9530a) y consulta este Genially para profundizar en ellos.

## **Producción y Programación**

**DISEÑO MOBILE FIRST**

El concepto de diseño mobile first implica plantear el proceso de diseño, teniendo en cuenta el teléfono en primer lugar. Esta forma de trabajo nos obliga a pensar en el teléfono como punto de partida, priorizando lo esencial de un producto, y haciendo foco sólo en lo que tiene sentido para este dispositivo.

Desarrollar para pantallas pequeñas nos ayuda a eliminar todo lo que no es estrictamente necesario para nuestro producto. Teniendo en cuenta que los dispositivos móviles representan más de la mitad del mercado digital se hace imprescindible este abordaje. Resulta más sencillo tener pocos elementos y funcionalidades e ir agregando más a medida que van creciendo los usuarios.

Algunas buenas prácticas en este punto son:

1. **Priorizar Contenido:** En Mobile-first el contenido es clave. Los elementos críticos deben visualizarse correctamente.
2. **Navegación intuitiva:** Es importante darle peso a los elementos que ayudan con la navegación.
3. **Sin Pop-Ups:** El espacio es limitado. Evitar ventanas flotantes que lo invadan.
4. **Testeo**: La mejor forma de asegurarnos que el diseño está optimizado es testeándolo con usuarios.

Recuerda la filosofía Customer Centricity. La clave es enfocarnos en lo que es importante para nuestros usuarios.

**LENGUAJES DE PROGRAMACIÓN**

Es el proceso de escribir instrucciones para la computadora, en el formato y sintaxis que el lenguaje de programación requiere, a fin de que se realice determinada acción. Programar es comunicar. En el universo de la programación, se suele hacer una diferenciación entre desarrollo front-end y back-end, según las funciones que cumple:

* Front-end: maquetación, lógica visible de un sitio web. Es lo que puede ver un cliente, y con lo que interactúa.
* Back-end: desarrollo de APIs del lado de los servidores y bases de datos. Se conecta la base de datos con la interfaz de la web/app.

**Principales lenguajes de Frontend:** JavaScript, HTML y CSS.

**Principales lenguajes de Backend:** Python, Java, Ruby, PHP.

**Nota**: HTML y CSS son lenguajes, pero no de programación. Sirven para dar estructura y estilo al sitio.

**FRAMEWORKS**

Por último, los frameworks (o marco de trabajo), son un conjunto de herramientas y clases que nos permiten solucionar un problema o funcionalidad. Están escritos en uno o más lenguajes de programación. Los frameworks establecen, además, una estructura determinada para el código y todos los archivos, y una metodología armónica del proyecto.



**DIFERENCIAS ENTRE WEB Y APP**

**SITIO WEB RESPONSIVE**

* HTML + JS + CSS.
* Se adapta a cualquier resolución.
* No se basa en un Sistema Operativo (SO).
* Siempre está en la web.
* Pueden aplicarse mejoras progresivas.

**APLICACIONES WEB**

* HTML + JS + CSS.
* Independiente al SO.
* Puede que te ofrezca interactuar con el navegador o sistema operativo.
* No requieren actualizarse, el usuario siempre está viendo la última versión.
* Requieren una conexión a internet.
* No aprovecha al máximo el hardware de los dispositivos.
* Ventajas: Las Web Apps pueden ofrecer a los usuarios el look and feel de las apps sin limitación de sistema operativo o dispositivo, además podemos brindar uso sin necesidad de instalarlas. Otra ventaja de su desarrollo es el de un costo menor en tiempo y recursos en comparación con las apps.

**APLICACIONES NATIVAS**

* Gestión desde la Tienda de Apps.
* Usan las funcionalidades del SO.
* Se desarrolla con el Software Development Kit (SDK).
* Se diseñan y programan para cada SO.
* No siempre requieren internet para funcionar.
* Permiten utilizar el hardware.
* Interfaz basada en las guías para cada SO para favorecer la usabilidad.

**APLICACIONES HÍBRIDAS**

* Combinación entre las dos anteriores.
* Una vez desarrollada la web se compila en una App Nativa.
* Permite tener un mismo código.
* Con librerías es posible acceder a las funciones del teléfono.
* Frameworks: React, Angular, Ionic, Phonegap, Apache Cordova.
* Dato curioso: El desarrollo de aplicaciones híbridas tiene una curva de aprendizaje mucho más suave que el desarrollo nativo. Para desarrollos nativos, se requiere tener conocimientos en lenguajes de programación como Java / Kotlin / XML, para Android, y Swift / Objective-C para iOS, además de las herramientas y particularidades específicas de cada plataforma.

**COMO PM ¿DEBO SABER DESARROLLAR TODO ESTO?**

No, no es necesario. Como PM sólo debes anticiparte a lo que necesitas de estas herramientas. Por ello te sugerimos conocerlas 😉 Ejemplos: ¿Cuáles son los principales features que deben tomarse en cuenta para el wireframe? ¿Cómo queremos que sea la experiencia de los usuarios? ¿Cuáles son sus objetivos en la navegación? ¿Precisaremos de API's? ¿Para qué momento? ¿Qué datos necesitamos recabar en nuestra data base? ¿Por qué? ¿Qué tipo de app o web necesitamos? ¿Por qué esta elección y no otra? Entre otras decisiones clave.

Sé estratégico: Consulta con tu Squad y aprovecha los conocimientos de tu equipo de desarrollo para que puedas tomar decisiones lo más acertadas posible en cada momento.

**¿Cómo seleccionar el tipo de desarrollo?**

Para elegir el tipo de plataforma que más nos conviene debemos tomar en cuenta ciertos factores:

* Tiempos y costos de desarrollo.
* Tipos de dispositivos donde queremos lanzar nuestro producto.
* Necesidad de acceso offline.
* Facilidades de mantenimiento.
* Ingresos publicitarios (En los sitios web los usuarios pueden usar adblockers lo cual puede limitar nuestros ingresos por publicidad)

Luego de esta clase, podrás empezar a optimizar tu proyecto final para la segunda pre-entrega, incorporando:

* Tipo de desarrollo: Web o App y tipología.
* Referencias visuales + Wireframes en baja o puntos importantes a tener en cuenta para la Arquitectura de la información. Piensa en Key Features y las User Stories definidas. Aquí podrás ayudarte con el Genially enlazado en la etapa de “Pre-producción”.
* Puedes ingresar aquí al [**template del brief final**](https://docs.google.com/presentation/d/1_VM0qdhFjvQYWi4rP2-RcL9TQ6fVtKZculNnAKcB5Y4/edit?usp=sharing) y tomarlo como referencia para optimizarlo con lo visto en clase 🚀. Te recomendamos continuar con tu PF para llegar con tiempo y de la mejor manera la segunda pre-entrega.

**Resumen de lo visto en la clase de hoy:**

* Vinculación entre PM & Desarrollo
* Pre-Producción
* Producción y programación
* Diferencias entre Web & App
* Lenguajes de programación
* Diseño Mobile First
* Wireframes, Api y Estructura de la base de datos.

#DemocratizandoLaEducación