

Pomodoro

Daniel E. Hernández

2019-05-16

Resumen

Este proyecto busca incrementar la productividad de los usuarios por medio del uso de la *Técnica Pomodoro*. Dicha técnica se ve plasmada en la función principal de una aplicación. Dicha función es un temporizador de 25 minutos y una función secundaria es poder romper en *tasks* las ideas del usuario y lo que él necesite hacer.

1. Descripción General

1.1. Problema

1. **Al tener una tarea grande, es difícil subdividir:** Al momento de tener una tarea principal en mano, muchas veces es difícil lograr realizarla en el tiempo indicado dado que no se tiene una idea específica de cada sub-tarea o *task* de qué hacer sino solo una idea abstracta general.
2. **Tener muchas tareas dificulta la segmentación de tiempo:** Al momento de tener muchas tareas principales y un tiempo limitado, muchas veces es difícil segmentar dichas tareas en rangos de tiempo. Por ende, es difícil lograr terminar la tarea principal en el tiempo máximo dado para ella (*o deadline*).
3. **Descansos irregulares = menos productividad y más cansancio:** Al no tener rangos de tiempo determinados para poder descansar mientras se está haciendo una tarea principal/sub-tarea resulta que en general se es menos productivo dado que al cansarse, se suele tomar descansos irregulares de amplia cantidad de tiempo.

2. Metas

1. **Al tener sub-tareas o *tasks* es más fácil cumplir con ellas:** Se logran cumplir porque se tiene una idea específica de cada objetivo que se quiere lograr en vez de solo tener una visión abstracta general.

2. **Completar la tarea principal o *objective* en *sprints* o *Pomodori* (*plural de Pomodoro*) pequeños:** Se logra cumplir el *objective* abstracto y general por medio de sprints donde el usuario eligió los tasks que quiere completar en el mismo. Cada Pomodoro es de 25 minutos según lo recomendado por la *Técnica Pomodoro*.
3. **Descansos de un intervalo de tiempo establecido o *Recess*:** Al final de cada Pomodoro el usuario descansa por 5 minutos o más según él elija, y al terminar su tiempo de descanso, inicia otro Pomodoro. Al lograr esto maximiza su productividad porque hay un descanso efectivo y no son lapsos de descanso demasiado grandes interrumpiendo el *train of thought* o el proceso productivo.

3. Fuera del enfoque

1. **GUI y UX intuitivo:** Tener una interfaz gráfica intuitiva y autoexplicativa.
2. **GUI según lenguaje de diseño de plataforma:** Seguir el diseño sugerido por Apple para iOS, utilizar *Swift* para que la aplicación corra en el nivel más bajo permitido y recomendado por Apple junto con *Objective-C* en casos sea necesario (i.e. según sea sugerido por la documentación de ciertos métodos de Swift como: `Timer()` que necesita un `#selector` que utiliza una función de Obj.-C)
- 3.

4. Personas y roles:

1. **Daniel Hernández (*dev y ops*):** Crear el proyecto, desarrollarlo, mantenerlo y publicarlo en el App Store.
2. **Fernando José Boiton y Juan Luis López (*QA*):** Revisiones de entrega del proyecto.

5. Contexto

5.1. Casos de Uso

5.1.1. El usuario quiere:

- **Ser más ordenado**
Quiere organizar sus ideas al ordenarlas en Objectives y Tasks.
- **Ser más eficiente**
Cumplir con los intervalos de tiempo diseñados por la Técnica Pomodoro.

- **Descansar y no procrastinar**

Al tener intervalos de tiempo determinados, se evita procrastinar y hay descanso efectivo.

6. Propuesta

6.1. Pomodoro App

Es una aplicación que busca incrementar la productividad y eficiencia del usuario por medio de la *Técnica Pomodoro* al ayudarlo a organizar sus ideas y dividir mejor su tiempo.

6.2. User Experience

1. **Timer:** de 25 minutos que cambia a 5 minutos una vez se acabe cada Pomodoro.
 - a) **Timer:** Inicia siempre con 25 minutos y se van restando minuto por minuto hasta llegar a 0.
 - 1) Caso Pomodoro: Inicia con 25 minutos.
 - 2) Caso *Recess*: Inicia con 5 minutos.
 - 3) Caso *Long Recess*: Inicia con 20 minutos.
 - b) **Botón de *Start*:** Al utilizarse se inicia o el Pomodoro, el *Recess* o el *Long Recess*.
 - c) **Botón de *Stop*:** Cancela o el Pomodoro, el *Recess* o el *Long Recess*.
 - d) **Botón de +5 minutos:** Al utilizarse le agrega a la actividad actual en el Timer 5 minutos con un límite máximo de 25 minutos (i.e. si el Pomodoro iba por el minuto 20 y presiona el botón de +5 minutos, aumentará el Timer a 25 minutos, empero, lo presionara una vez más no agregaría más tiempo puesto que ya se encuentra en el límite máximo de 25 minutos).
 - e) **Label: *Currently*:** Indica cualquiera de las tres posibles actividades en las que se podría encontrar el usuario: Pomodoro, *Recess* o *Long Recess*.
 - f) **Label: *Pomodori*:** Indica la cantidad de veces que se ha realizado un Pomodoro, eso es, cada vez que el *Label: Currently* indica que el usuario está dentro de un Pomodoro y el Timer se resta desde 25 minutos hasta 0, el *Label: Pomodori* indica la cantidad que ya tenía +1.
 - g) **Label: *Sessions*:** Luego de completar 5 *Pomodori*, se completa una sesión y el label muestra la cantidad que ya tenía de sesiones +1.
2. **Task Managment:**

- a) **Screen: *All Tasks*** Donde cada task tiene su propia celda y la celda tiene sus atributos:
- 1) **Task Name** Indica el nombre específico del task en la celda.
 - 2) **Task Rating o Difficulty** Indica qué tan difícil es de realizar ese task o la prioridad según la cantidad de estrellas que el usuario elija. La interpretación de qué significan las estrellas varía según el usuario pero las dos definiciones propuestas son: Dificultad o Prioridad.
 - 3) **Objective** Indica a qué *objective* pertenece, eso es, a qué task abstracto o general pertenece este task que es específico.
 - 4) **Add Button** Siguiendo los *guidelines* de Apple, el botón de + pasa del screen: *All Tasks* a el screen: *Add Task*.
- b) **Screen *Add Task*** Donde el usuario puede ingresar los datos específicos del task y si los deja en blanco (gracias a el uso de *Optinals* en *Swift*) toman un setting o dato *default*. Contiene los siguientes campos para ingresar dichos datos que luego serán utilizados por el screen *All Tasks*.
- 1) **Botón de *Cancel***: Inicializa un *segue* para regresar al screen: *All Tasks* sin guardar los cambios efectuados.
 - 2) **Botón de *Done***: Inicializa un *segue* para regresar al screen: *All Tasks* guardando los cambios efectuados.
 - 3) **Text Field: TaskName** Donde el usuario ingresa el nombre de su task.
 - 4) **Detail View: Objective Name** Al seleccionarlo se inciaa un *segue* a la siguiente screen de *Choose Objective* donde el usuario elige el *Objective* de su task.
 - 5) **Text Field: Difficulty** El usuario ingresa un número recomendado entre 1 – 5 que definirá la cantidad de estrellas asignadas a ese task específico. Si el número fuera ≤ 0 entonces la cantidad asignada es 1 estrella y si fuera ≥ 5 la cantidad asignada es 5 estrellas.

6.3. Trabajo a futuro:

1. ~~**Mejor diseño:** Diseñar mockups de la interfaz gráfica.~~
2. ~~**iOS App:** Desarrollar la primera versión de la aplicación.~~
3. **Implementación de B-Tree:** Añadir B-Tree para guardar los structs, en vez de array de structs.
4. **Pruebas de usuario:** Realizarlas para asegurar el uso intuitivo de la aplicación y corregir cualquier falla.
5. **Mark as done:** Add checkbox (or equivalent gesture) to tasks to be able to mark them as Done.

6. **Add Objectives:** Enable the user to add Objectives.
7. **Sample Data:** Eliminar el sample data para tener la aplicación limpia para la personalización del usuario.

7. Tasks y Deadlines

1. **PRD:** 2019-02-28
2. **MVP:** 2019-04-23
3. **Final:** 2019-05-13 – 2019-05-17