



CC5901- PRÁCTICA PROFESIONAL II

---

# Informe de Práctica Profesional

## *Solar Tracker Reckoner*

---



Ignacio Slater M.

19.133.399-3

ignacio.slater@ug.uchile.cl

(+56) 9 9158 5187

**Duración de la práctica:**

19/01/2020 - 28/02/2020

11 de abril de 2020  
Santiago, Chile

## Resumen

En el presente informe se explica el planteamiento y la solución de un proyecto de *software* de manejo de flotas bajo el contexto de práctica profesional de ingeniería civil en computación.

La práctica se llevó a cabo en la célula *Ocean*, perteneciente a *Entel* y tuvo una duración aproximada de dos meses a jornada completa.

El objetivo final de la práctica era el de entregar un producto mínimo viable del sistema solicitado. La solución planteada fue una aplicación móvil basada en una arquitectura *serverless* modelada como cliente-servidor.

La modalidad de trabajo se organizó en *sprints* de una y dos semanas, y se rigió por principios de agilidad, donde al final de cada *sprint* se presentaban los avances del proyecto.

El sistema entregado al terminar del período de práctica consistió en dos aplicaciones, una para administradores del sistema, y otra para los conductores de los vehículos monitoreados. Ambas aplicaciones fueron desarrolladas con un *backend* en *Python* y un *frontend* en *ReactNative*.

## Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Descripción de la empresa . . . . .	2
1.2. Motivación de la empresa . . . . .	2
1.3. Descripción general del trabajo realizado . . . . .	2
<b>2. Problema abordado</b>	<b>3</b>
<b>3. Objetivos</b>	<b>4</b>
<b>4. Metodologías</b>	<b>5</b>
<b>5. Descripción de la solución</b>	<b>6</b>
5.1. Aplicación del administrador . . . . .	6
5.2. Aplicación del conductor . . . . .	8
5.3. Problemáticas encontradas . . . . .	8
<b>6. Discusión y reflexión sobre la práctica</b>	<b>9</b>
<b>7. Conclusiones</b>	<b>11</b>
<b>Bibliografías</b>	<b>12</b>

## 1. Introducción

### 1.1. Descripción de la empresa

La práctica profesional se realizó en *SolarTracker*, un *startup* ubicado en el sector de Providencia, Santiago.

La alineación de la empresa se basa en tres ejes fundamentales:<sup>1</sup>

1. Potenciar la generación de energías renovables para el desarrollo sostenible de las necesidades industriales y sociales a nivel global.
2. Facilitar el uso de la energía solar, optimizando la operación y mantenimiento mediante monitoreo inteligente y predicción de eventos de forma automática.
3. Potenciar el desarrollo de energías fotovoltaicas en Chile y LATAM, simplificando las capacidades técnicas de instalación y operación.

### 1.2. Motivación de la empresa

Al ser un *startup* pequeño, con poco tiempo en el rubro y relativamente poca experiencia y presupuesto, la empresa decide buscar practicantes para apoyar al equipo de desarrollo a implementar nuevas herramientas al proyecto principal de la compañía.

En total se buscaban dos practicantes: un estudiante de ingeniería civil eléctrica para realizar un trabajo de investigación para impulsar el futuro desarrollo de la plataforma, y un estudiante de ingeniería civil en computación (autor del presente informe) para adaptar el software existente a nuevos estándares y tecnologías.

### 1.3. Descripción general del trabajo realizado

Como ya se mencionó, el objetivo de la práctica era adaptar una plataforma existente para utilizar nuevas tecnologías. En particular, migrar una base de datos a un nuevo esquema para utilizar las tecnologías de *BigQuery*,<sup>2</sup> considerando todos los cambios que esta migración tenía sobre la plataforma ya construida.

Las tecnologías utilizadas para resolver el problema fueron el lenguaje de programación *Python*,<sup>3</sup> *MongoDB*,<sup>4</sup> *DAGs* de *Apache Airflow*<sup>5</sup> y la ya mencionada *BigQuery*.

---

<sup>1</sup> Sitio web de Solartracker.

<sup>2</sup> Sitio web de BigQuery.

<sup>3</sup> Sitio web de Python.

<sup>4</sup> Sitio web de MongoDB.

<sup>5</sup> Directed Acyclic Graphs.

## 2. Problema abordado

Este cambio implicaba principalmente tres problemáticas que debían ser abordadas:

1. Se debían migrar los datos del antiguo motor de bases de datos<sup>6</sup> al nuevo.
2. Se necesitaba adaptar

Dada la gran demanda de servicios por parte de empresas a *Entel*, y el aumento de competidores en el mercado de telefonía móvil, se vio una oportunidad de crear una célula de investigación y desarrollo de *software* dentro de la empresa.

El proyecto de creación de una aplicación de manejo de flotas surge por una necesidad de los clientes de la empresa para manejar el estado de sus vehículos. Al comenzar la práctica *Entel* ya contaba con un sistema que otorgaba este servicio, pero no era propietario de la empresa, así que se requería implementar un sistema similar desarrollado por el nuevo equipo de desarrollo.

El problema principal fue el de crear dicha aplicación para poder ofrecerla a los usuarios finales para el mes de abril de 2019.

---

<sup>6</sup>*Firebase*.

### 3. Objetivos

El objetivo principal de la práctica fue crear una plataforma que permitiera a empresas de pequeño y mediano tamaño manejar y administrar flotas de vehículos de manera centralizada. Además la aplicación debía contar con dos tipos de usuarios: administradores y conductores.

Los administradores estaban encargados de monitorear el estado de los vehículos y las entregas encargadas a la empresa e informar de cualquier problema que surgiera durante su traslado.

Por otro lado, la aplicación de los conductores debía enviar la ubicación del vehículo durante toda la ruta e informar de cualquier cambio de estado del pedido. Adicionalmente, se requería que tuviera integración con el *GPS* del dispositivo móvil y que pudiera sugerir una ruta óptima de entrega para el conductor (considerando que una ruta está compuesta de una cantidad arbitraria de encargos).

El sistema a desarrollar fue pensado como una aplicación móvil dado la versatilidad de esta plataforma y la comodidad que le brindaría a los usuarios.

El sistema se planteó con una arquitectura *serverless* que utilizara los servicios de *AWS*<sup>7</sup>. Junto con lo anterior, se limitaron las herramientas de desarrollo de la aplicación a *Python* y *React Native*.

El objetivo final esperado al terminar el período de práctica era la entrega de un MVP<sup>8</sup> que cumpliera con los requisitos mínimos para que la aplicación brindara valor a sus usuarios.

---

<sup>7</sup>*Amazon Web Services*

<sup>8</sup>Producto mínimo viable

## 4. Metodologías

Para solucionar el problema planteado se siguió una metodología de investigación y desarrollo. El proceso de investigación fue adoptado ya que gran parte de el equipo desconocía las herramientas utilizadas para implementar el producto.

Además, el trabajo fue organizado siguiendo principios ágiles de desarrollo. El esquema de desarrollo planteado estaba compuesto de las siguientes etapas:

- *Stand up meetings* diarios al finalizar el día de trabajo para visualizar el avance del proyecto y detectar errores de manera temprana.
- *Sprints* de dos semanas dirigidos por una *Scrum Master* con objetivos que eran revisados al final del *sprint* en un *review*.
- Reuniones semanales con el vicepresidente del área de corporaciones de la empresa para informar de los avances de los proyectos de la célula y presentar oportunidades, ideas y posibles ofertas.

Adicionalmente, siguiendo también principios ágiles se utilizó un kanban para organizar el trabajo. El kanban contenía *historias de usuario* que representaban hitos en el desarrollo del sistema. Luego, cada historia de usuario se dividía en tareas atómicas en las que podía trabajar uno o más de los integrantes del equipo dependiendo de su dificultad. Todas estas tareas pasaban por una revisión de calidad antes de considerarse terminadas.

Uno de los principales beneficios de seguir las metodologías mencionadas fue la flexibilidad que otorgaba al momento de organizar y fijar los objetivos del proyecto, ayudando a aclarar el trabajo que debía realizarse en cada jornada.

Las metodologías utilizadas probaron ser efectivas, obteniéndose los resultados presentados en la siguiente sección.

## 5. Descripción de la solución

La solución al problema planteado se modeló finalmente como dos aplicaciones móviles, una para el administrador del sistema y otra para los conductores.

La aplicación del administrador le permitía manejar una pequeña flota de vehículos y asignar tareas a cada uno de estos.

Por otro lado, para los conductores el programa mostraba una visualización de las tareas encomendadas por el administrador (mostrando la ruta hasta destino y los detalles del pedido) que debía marcar como terminado una vez entregado. Además se le daba la opción a este usuario de aceptar o rechazar una tarea.

Para implantar la solución se utilizó el *framework serverless.com* que permitía una fácil configuración para distintos servicios de *cloud computing*, entre los cuales se encuentra AWS mediante uno o varios archivos *YAML*.

A continuación se presentan las funcionalidades y detalles de la implementación para estas aplicaciones.

### 5.1. Aplicación del administrador

Para este proyecto, un administrador es un usuario que maneja a los conductores de una empresa. Ellos se encargan de:

- Crear y asignar tareas a conductores
- crear nuevos conductores y nuevos vehículos

Con el fin de facilitar el flujo de información de todos los vehículos, se diseñaron cuatro pestañas en la parte inferior de la aplicación.

1. La primera pestaña comprendía la información de todos los conductores pertenecientes al sistema. La implementación se hizo de tal forma que al presionar el nombre de alguno de los conductores se desplegara el detalle de su estado, mostrando su nombre, ubicación, tareas pendientes y el vehículo en el que se trasladaba.
2. En la segunda pestaña se podía acceder a los detalles de los vehículos. La información contenida en esta sección es análoga a la del conductor.
3. La tercera pestaña contenía la información de las tareas creadas por el administrador. En esta sección el administrador tenía la capacidad de crear tareas nuevas y asignarlas a conductores y vehículos.
4. La última pestaña contenía reportes de uso de la aplicación y los vehículos, tales como la cantidad recorrida por estos el último mes, la cantidad de tareas completadas por conductor, entre otras.



Para almacenar y administrar los datos se utilizaron bases de datos no relacionales, específicamente el motor *DynamoDB* ofrecido por *AWS* dada la flexibilidad que ofrecía respecto a utilizar bases de datos relacionales y su fácil integración con servicios *serverless*.

Para conectar la aplicación con el sistema de *cloud computing* se utilizó el servicio *AWS API Gateway* que permitía la comunicación entre las distintas partes del programa con los servicios de *Amazon*.

La conexión con la base de datos y el manejo de la información de estas se realizó en *Python*, modelando las *requests* a la *API* mediante el uso de *AWS Lambda*. Así, cada petición se estructuraba como una sola función *lambda*, lo que evitaba tener que manejar servidores, facilitando el modelo de computación en la nube de la aplicación. Además, esto último también facilitaba la integración de plataformas de *IoT* al sistema.

El acceso a la base de datos por parte de *Python* se hacía mediante la librería *boto3* distribuida por *Amazon*.

Para diseñar el *frontend* de la *app* se utilizó *React Native*, lo que permitía crear aplicaciones nativas para plataformas móviles utilizando *JavaScript*. De esta forma, no eran necesarias implementaciones distintas para *iOS* y *Android*, ya que el código era compilado a una aplicación nativa para cada una de estas plataformas.

El diseño de la interfaz gráfica fue destinada a especialistas en el área de *UX*, quienes optaron por seguir las normativas de diseño de *Google*, en particular las definidas en *Material Design*. Luego, el equipo de desarrollo debía implementar las vistas diseñadas en *React Native*, lo que fue una de las problemáticas más importantes a la hora de poner en funcionamiento la plataforma.

Si bien *React Native* proveía de las funcionalidades básicas para implementar aplicaciones nativas, la mayoría de las herramientas estaban construidas por la comunidad por lo que carecían de soporte oficial. Esto último llevó a que varias de las decisiones de diseño tomadas en un comienzo tuvieran que modificarse para adaptarse a las limitaciones del *framework*.

Entre las herramientas de *React* utilizadas destacan: *react-native-paper*, que provee los elementos necesarios para desarrollar la interfaz de usuario siguiendo los lineamientos de *Material Design*; *react-native-maps*, que permitía una conexión de la aplicación con los servicios de *Google Maps* y *Google Places*; y las librerías correspondientes a los servicios de *AWS* como *Amplify* y *Cognito*.

La conexión entre el *backend* y el *frontend* de la aplicación se hizo mediante los *endpoints* generados por *AWS API Gateway*.

## 5.2. Aplicación del conductor

La aplicación por parte del conductor se implementó de manera similar a la del administrador, utilizando los servicios de *AWS* junto con *Python* y *React Native* para el *backend* y *frontend* respectivamente.

La gran diferencia de este sistema con el presentada en la sección anterior es que el conductor tiene menos herramientas disponibles.

Entre las facultades que posee el conductor están las de aceptar y rechazar encargos, marcar algún encargo realizado como terminado y cambiar su estado a disponible o inactivo. Además, al momento de aceptar un pedido, la aplicación decide una ruta óptima hasta el destino utilizando la *API* de *Google Maps*.

## 5.3. Problemáticas encontradas

El problema más importante al momento de desarrollar la solución fue lo acotado de los tiempos de entrega, debido a los plazos que se tenían y a que muchas veces los días laborales se veían acortados por reuniones.

Además, como se mencionó previamente, uno de los mayores desafíos de la implementación fue el diseño de la interfaz gráfica con *React Native*. Las principales causas de esta dificultad se debieron al desconocimiento del equipo respecto a estas herramientas. Lo anterior implicó dedicar gran parte del tiempo a conocer los estándares de desarrollo, las herramientas de *testing* y de *debugging*.

## 6. Discusión y reflexión sobre la práctica

Un primer aspecto a mencionar de la práctica es que en un comienzo estaba pensada para que los practicantes trabajaran en 2 proyectos: el **sistema de manejo de flotas** del que se habla en el presente informe, y un **sistema para detectar episodios de epilepsia** mediante la lectura y análisis de ondas cerebrales mediante el uso de *machine learning*. Finalmente solamente se trabajó en el primer proyecto mencionado debido a lo acotado de los tiempos y ya que le generaba un mayor valor a la empresa.

Uno de los primeros problemas a los que se vio enfrentado el equipo fue la inexperiencia de todos sus miembros, dado que se trabajó con tecnologías muy nuevas (e.g. *React Native* fue liberado en marzo del 2015, encontrándose aún en la versión 0.6). A lo anterior se suma que en un comienzo el equipo de desarrollo estaba formado por 4 practicantes y un solo profesional (el supervisor de los practicantes)<sup>9</sup> y que no existía ninguna base sobre la que trabajar puesto que el proyecto aún no había sido puesto en marcha, encomendándose esta tarea a los practicantes.

Adicionalmente, las primeras semanas de la práctica sirvieron como un proceso de nivelación dado que no todos los miembros del equipo habían recibido la misma formación.

Otro problema relacionado a lo acotado de los plazos de entrega fue que perjudicó el trabajo al momento de plantear herramientas y modalidades alternativas para el desarrollo de la aplicación, teniendo muchas veces que quedarse con herramientas subóptimas dado que no había espacio para investigar sobre otras. Sin embargo, este problema pudo ser abordado de buena manera gracias a las metodologías ágiles adoptadas por el equipo, ya que permitieron una mejor comunicación y organización del equipo.

Una fortaleza que vale la pena mencionar es que la formación de los practicantes, (en particular los pertenecientes a la especialidad de computación) entregaba gran parte de los conocimientos necesarios para facilitar el aprendizaje de las tecnologías a utilizar, además de haber desarrollado una mentalidad crítica para lograr plantear y discutir los posibles diseños para solucionar los diversos problemas a resolver. Entre los cursos que fueron de utilidad para lo anterior destacan:

- *Metodologías de diseño y programación*, que permitió concretizar soluciones con un buen diseño;
- *Bases de datos*, que ayudó a organizar y comprender los datos que debía manejar el sistema;
- *Ingeniería de software*, que sirvió de experiencia previa para trabajar en equipo y adoptar herramientas de desarrollo con un modelo *cliente-servidor*, además de servir de introducción a principios ágiles; y

---

<sup>9</sup>En los días siguientes se sumaría una nueva practicante al equipo y otros profesionales, pero el equipo de desarrollo seguía formado en su mayoría por estudiantes.

- *Lenguajes de programación*, que simplificó la tarea de adaptarse a las nuevas herramientas utilizadas durante el período.

En esa misma línea, también se pudo notar que el haber cursado los cursos de especialidad facilitó que los miembros profesionales del equipo tomaran en cuenta las ideas propuestas por los practicantes.

Si lugar a dudas, uno de los mayores problemas al momento de trabajar en la práctica fue la falta de conocimientos previos (como se ha mencionado anteriormente) acerca de las tecnologías empleadas, y habría hecho más fluido el avance del proyecto si dichos conocimientos hubieran existido antes de comenzar.

A pesar de todas las dificultades encontradas, la buena comunicación del equipo y el apoyo, tanto de los otros practicantes como de los trabajadores de la empresa, ayudó inmensamente a que siempre lograran encontrarse soluciones a las problemáticas que surgieran.

## 7. Conclusiones

Como se mencionó en secciones anteriores, el objetivo principal de la práctica fue desarrollar una primera versión del sistema de manejo de flotas y concretar un producto mínimo viable. Este producto consistía de las dos aplicaciones descritas previamente (una para el administrador y otra para los conductores), con una interfaz de usuario nativa para *Android* y un servicio *serverless* montado en la nube de *AWS* para manejar la base de datos del programa. Al final del período de práctica, se logró cumplir con este objetivo, faltando solamente algunas funcionalidades de menor importancia (como que la aplicación pudiera tomar fotografías)

Los principales conocimientos obtenidos tienen que ver con las tecnologías utilizadas. Se aprendió a trabajar con servicios *serverless* y a desarrollar aplicaciones móviles utilizando *React Native*. Adicionalmente se aprendieron mecánicas de trabajo en equipo, de organización de tareas para trabajar en proyectos y de principios de desarrollo ágil.

Se concluye finalmente que la práctica fue una experiencia fructífera para desarrollar habilidades y aprendizajes relacionados al mundo laboral de ingeniería civil en computación, destacándose las competencias mencionadas en ésta y las secciones anteriores del informe.

## Bibliografías

---

**Sitio web de BigQuery****bigquery**

*BigQuery: Cloud Data Warehouse / Google Cloud.* URL: <https://cloud.google.com/bigquery> (visitado 07-04-2020).

---

**Directed Acyclic Graphs****dag**

*Concepts — Airflow Documentation.* URL: <https://airflow.apache.org/docs/stable/concepts.html> (visitado 11-04-2020).

---

**Firebase****firebase**

*Firebase.* URL: <https://firebase.google.com> (visitado 11-04-2020).

---

**Sitio web de Solartracker****solar-tracker**

*Solartracker: Servicio de Monitoreo de plantas fotovoltaicas basado en Inteligencia artificial.* URL: <https://www.solartracker.cl> (visitado 07-04-2020).

---

**Sitio web de MongoDB****mongo**

*The most popular database for modern apps / MongoDB.* URL: <https://www.mongodb.com> (visitado 11-04-2020).

---

**Sitio web de Python****python**

*Welcome to Python.org.* URL: <https://www.python.org> (visitado 11-04-2020).