

# Marcas Tracker

*Camila Campagnolo, Victoria Casaravilla, Linda Gareis*

**Palabras clave:** shiny, marcas, reportes, detección de errores

El siguiente working project<sup>1</sup> tiene como objetivo automatizar el cómputo de horas trabajadas de las empresas tercerizadas en una organización. Se procesan los registros de las marcas de entradas y salidas de funcionarios tercerizados y se genera una aplicación en la que se visualiza e interactúa con diferentes reportes de control. Como resultado, de manera automática la aplicación calcula las horas trabajadas y detecta inconsistencias en los registros.

## 1. Introducción

Este proyecto surge en un contexto de transformación digital a partir de la necesidad de automatizar un proceso de control y facturación de horas de funcionarios. Este trabajo solía realizarse manualmente generando ineficiencias y errores operativos.

A modo de solución, a través del uso de R, se propone la generación de una aplicación con Shiny de manera que se haga el procesamiento completo de los datos y que genere una visualización de los reportes requeridos.

Con este objetivo, se comienza identificando las inconsistencias en los registros y realizando un reporte de las mismas. Dicho reporte incluye un detalle de cada tipo de potencial error en las marcas. De esta manera, es posible identificar fácilmente las incidencias y sustituir la marca reportada como errónea por la marca que debería haber sido ingresada.

Además, se realizan dos reportes adicionales que muestran las horas trabajadas de manera agregada como desagregada por horario, feriados, descanso y extras. La aplicación muestra el cálculo de horas pero identifica los casos en los cuales es posible que existan errores debido a la presencia de inconsistencias.

El producto propuesto resultó en un ahorro de horas dedicadas a buscar inconsistencias y realizar cómputo de horas de forma manual.

## Aspectos técnicos

El objetivo principal consistía en minimizar el trabajo manual tanto del control y cálculo de horas como de ajustes del código debido a que los encargados de trabajar con la aplicación aún no tienen un conocimiento en profundidad de R.

Por esta razón, el código no requiere ninguna actualización manual, para utilizarlo solo se debe guardar los datos del mes en la carpeta del código y correr la aplicación. Esta adaptabilidad del working directory es posible debido a la incorporación de la librería **here**

---

<sup>1</sup> El proyecto aún se encuentra en proceso de desarrollo y mejoras.

que permite manejar rutas relativas dentro del proyecto evitando las restricciones en reproducibilidad de las rutas absolutas.

El código es capaz de adaptarse también a los cambios en el nombre del archivo input que hacen referencia al mes de los datos utilizando como base la fecha del sistema. Para esta función y la manipulación de fechas y horas de los datos se utiliza la librería **lubridate**.

Además, se trabaja con dos excels, el que contiene los nuevos datos del mes y uno con datos referenciales útiles para identificar códigos de marcas y tipos de contratos de las empresas. La utilización de **readxl** resulta indispensable para el acceso a estas bases de datos. Además, los usuarios pueden hacer cambios a esta base de datos referenciales o descargar reportes en excel directamente desde la aplicación y para ello hacemos uso también de las librerías **openxlsx** y **writexl**.

Como se mencionó anteriormente, el código realiza el procesamiento total de los datos para lo que es necesario el uso de la librería **dplyr**. Esta librería permite que se realicen los cálculos de los diferentes tipos de horas y la identificación de inconsistencias. Además, las inconsistencias son personalizadas según el tipo de contrato de cada empresa que determina por ejemplo los horarios de inicio y fin de horas nocturnas. De esta manera se obtiene un control personalizado. Cabe resaltar que esto no sería posible sin la incorporación de estructuras de control y operadores lógicos.

Por otra parte, se realiza web scraping con la librería **rvest** con el objetivo de extraer las fechas de feriados de una página web debido a que es necesario tener las fechas actualizadas automáticamente para poder generar las horas trabajadas en esos días.

Finalmente, elegimos utilizar la librería **Shiny** y **shinythemes** porque permite una mejor interacción del usuario con los reportes debido a sus pestañas, filtros e interfaz. Esta interacción con los reportes es posible debido a la librería **DT**. De esta manera, es posible la visualización intuitiva de las horas de cada una de las empresas y funcionarios. El usuario podrá además actualizar los datos de nuevas empresas o nuevos códigos de marcas y de esta manera personalizar aún más el análisis.

### Potencialidades

Como se tiene como insumo el horario que los funcionarios deberían cumplir por contrato, se podría dar un paso más y eliminar la necesidad de marcar las horas extra, lo que actualmente resulta en tiempo perdido, en tanto los funcionarios deben dejar de hacer sus tareas para ir a marcar la hora extra. Esta solución impactaría en el ahorro de tiempo, tanto para los funcionarios que deben controlar las marcas como de los empleados que marcan.

Por otra parte, se está trabajando en que los usuarios sean capaces de corregir las incidencias directamente desde la aplicación para que no se tenga que realizar ninguna corrección manual en la base de registros original.

La aplicación se encuentra disponible en el siguiente enlace:

<https://marcastracker.shinyapps.io/marcastracker/>