

**SOCIEDAD CHILENA DE INGENIERÍA HIDRÁULICA**  
**XXVII CONGRESO CHILENO DE INGENIERÍA HIDRÁULICA**

**MODELO HIDROLÓGICO SEMI DISTRIBUIDO EN PYTHON PARA CUENCAS  
NIVOGLACIARES**

**MATÍAS RIVEROS HIDALGO.<sup>1</sup>**  
**ALVARO OSSANDON ALVAREZ.<sup>2</sup>**  
**LINA CASTRO.<sup>3</sup>**

**RESUMEN**

En este trabajo se implementa el modelo glacio-hidrológico de Sanllehi (2023) en Python, lenguaje de programación de acceso libre. El modelo fue desarrollado originalmente en Matlab e integra tres submodelos: uno para el derretimiento nival (UEB; Tarboton et al., 1996), otro para el derretimiento glaciar (ETI; Pellicciotti et al., 2005), ambos de carácter distribuido, y un tercero para la escorrentía y simulación de caudales (Miguelés, 1994), de tipo semi-distribuido. La validación se realizó en la cuenca “río Juncal en Juncal”, zona central de Chile, comparando resultados numéricos y comportamiento de las simulaciones en ambos lenguajes para cada submodelo. Posteriormente, se calibró nuevamente el modelo en Python, obteniendo un nuevo set de parámetros que se pueda ajustar a los datos observados. La calibración implementada, se realizó de forma simultánea para los submodelos UEB y de escorrentía, dejando el submodelo ETI con parámetros fijos calibrados previamente. De esta forma se da prioridad a la simulación de caudales, alcanzando valores de métricas de eficiencia aceptables para esta variable ( $KGE_{cal} = 0.75$ ;  $KGE_{val} = 0.67$ ). En conclusión, se considera la implementación del modelo en Python como exitosa, aunque con diferencias numéricas entre los lenguajes de programación que se vieron compensadas mediante la recalibración. El costo computacional surgió como una limitación relevante, a la hora de calibrar y simular, pero se presenta un gran potencial de mejora y optimización del programa.

---

<sup>1</sup> Estudiante, Ingeniería Civil, Universidad Santa María – email: [matias.riverosh@sansano.usm.cl](mailto:matias.riverosh@sansano.usm.cl)

<sup>2</sup> Profesor Asistente, Departamento de Obras Civiles, Universidad Santa María - email: [alvaro.ossandon@usm.cl](mailto:alvaro.ossandon@usm.cl)

<sup>3</sup> Profesor Asistente, Departamento de Obras Civiles, Universidad Santa María – email: [lina.castro@usm.cl](mailto:lina.castro@usm.cl)

## 1. INTRODUCCION

La disponibilidad de agua en la zona central de Chile depende en gran medida del derretimiento de la nieve acumulada en la cordillera de los Andes, así también como del deshielo de sus glaciares (Ayala et al., 2020). Sin embargo, desde 2010, la zona central de Chile ha enfrentado una mega sequía con déficits medios de precipitación de hasta un 40% (Garreaud et al., 2020). En esta región, la cuenca del río Maipo destaca por albergar más de 800 glaciares que desempeñan un papel crucial en la mitigación de los efectos de esta sequía (Ayala et al., 2020), que en los últimos años ha provocado un retroceso acelerado de los glaciares y una disminución de la precipitación tanto líquida como sólida. Como resultado, los recursos hídricos han disminuido drásticamente, con déficits máximos del 70% en los caudales medios de las cuencas de la región central y, en algunos casos, pérdidas totales de hasta el 100% en ciertos embalses precordilleranos de la misma zona (CR2, 2015).

Esta situación genera la necesidad urgente de encontrar técnicas efectivas para estimar los recursos hídricos en momentos de crisis. En este contexto, los modelos hidrológicos se han convertido en herramientas fundamentales para cuantificar la disponibilidad hídrica, permitiendo analizar los procesos glacio-hidrológicos de una cuenca considerando la variabilidad espacial y temporal (Van Tiel et al., 2020). Estos modelos buscan representar los procesos físicos del ciclo hidrológico mediante formulaciones matemáticas idealizadas, cuya complejidad puede variar según el grado de detalle con el que se quiere representar la realidad de la cuenca, las variables incluidas y la discretización espacial. Por ello, disponer de un programa de código abierto otorga versatilidad al modelador, permitiendo incorporar distintos procesos y ajustar el modelo a las características específicas de cada cuenca o dar un enfoque particular al fenómeno hidrológico que se desee estudiar.

Por lo tanto, el objetivo de este trabajo es **implementar un modelo hidrológico en Python**, cuya última versión fue desarrollada originalmente en MATLAB por Sanllehi (2023) e integra 3 módulos con diferentes submodelos: el modelo UEB (Utah Energy Balance; Tarboton et al., 1996), que calcula el aporte nivo-glacial de forma distribuida sobre la cuenca; el modelo ETI (Enhanced temperature-index; Pellicciotti et al., 2005), para estimar el derretimiento glaciar, también de carácter distribuido; y el modelo de escorrentía de Migueles (1994), que aborda los flujos superficiales y subterráneos de forma semi-distribuida, dividiendo la cuenca en bandas. La elección de este nuevo lenguaje permitirá que el programa sea universal y accesible para quienes deseen utilizarlo y/o complementarlo, creando una herramienta útil para cuantificar y evaluar la variación del recurso hídrico en escenarios futuros, especialmente bajo el contexto del cambio climático. El modelo se aplicó en la subcuenca “Río Juncal antes junta Estero Juncalillo”, de la cuenca del Río Aconcagua, Zona Central de Chile. Se compararon los resultados con el modelo original validando su replicabilidad en el nuevo lenguaje de programación y una nueva calibración demostró su versatilidad para trabajos futuros.

## 2. MARCO TEORICO

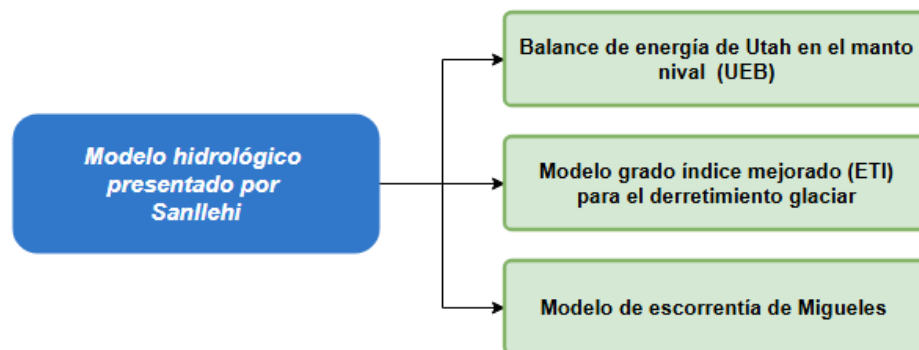
### 2.1. MODELO HIDROLÓGICO USM MODIFICADO POR SANLLEHI

El modelo hidrológico USM fue originalmente desarrollado considerando dos módulos principales: un submodelo de acumulación y derretimiento nival semi-distribuido basado en balance radiativo (Stowhas, 1975), y un submodelo de escorrentía desarrollado por Migueles

(1994), también de carácter semi-distribuido. Posteriormente, Sanllehi (2023) implementó modificaciones al modelo, incorporando un nuevo esquema de derretimiento y acumulación de nieve con el objetivo de mejorar la representación física de los procesos nivales. En concreto, el módulo de Stowhas (1975) fue reemplazado por el modelo UEB (Utah Energy Balance; Tarboton et al., 1996), el cual opera de manera completamente distribuida en el espacio. Además, se integró un nuevo submodelo de derretimiento glaciar, del cual carece la versión original del modelo hidrológico USM, basado en el modelo ETI (Enhanced Temperature Index; Pellicciotti et al., 2005), de naturaleza conceptual, pero con fundamento físico y también distribuido espacialmente.

Estas incorporaciones transforman el modelo USM en un sistema glacio-hidrológico, aunque su estructura general mantiene un carácter semi-distribuido debido al módulo de escorrentía de Migueles (1994). Este submodelo realiza el balance de humedad del suelo considerando procesos como infiltración, evapotranspiración y ruteo del flujo superficial y subterráneo dentro de la cuenca. También considera almacenamientos tipo estanque que reciben el derretimiento generado por los módulos nival UEB y glaciar ETI, además de la precipitación líquida en zonas sin cobertura de nieve. Para su correcto funcionamiento, el modelo requiere discretizar la cuenca en bandas de elevación, por lo que los volúmenes de agua derivados del derretimiento nival y glaciar se agrupan en dichas bandas antes de incorporarse al submodelo de escorrentía como variables de entrada

A continuación, se presenta la Figura 1 que muestra como está construido el modelo de Sanllehi (2023), que estructuralmente consta de 3 módulos principales. Cada submodelo tiene su propio modulo y pueden trabajar de forma independiente si se les entregan las variables de entrada necesaria. Sin embargo, para representar el ciclo hidrológico de una cuenca, se debe seguir la siguiente secuencia: primero se debe simular el derretimiento nival y la cobertura de nieve en el módulo correspondiente al submodelo UEB, luego estas variables entran al módulo donde esta implementado el submodelo ETI para obtener el derretimiento glaciar, en caso de que la cuenca presente glaciares. Finalmente, las variables de salida de los submodelos anteriores se agrupan por banda para entrar en el módulo correspondiente al submodelo de escorrentía y obtener caudales en el punto de aforo de la cuenca.



**Figura 1.** Modelo glacio hidrológico propuesto por Sanllehi (2023).

## 2.2. BALANCE DE ENERGÍA DE UTAH (UEB) SUBMODELO NIVAL

El modelo “Utah energy balance” (UEB; Tarboton et al., 1996), es un modelo hidrológico de base física y de carácter distribuido a nivel espacial con resolución temporal horaria. Este

modelo simula la acumulación y derretimiento nival utilizando una capa única para representar el manto de nieve sobre la cuenca. Para esto se realiza un balance de energía radiativa y un balance de masa, donde se simula el comportamiento de dicha capa llevando un seguimiento de dos variables de estado principales, el equivalente de aguanieve (SWE) y la energía almacenada en el manto nival (U). Se define una temperatura de referencia para la definición del estado del agua en la fase de hielo a 0 [°C]. Las ecuaciones (1) y (2) representan la variación en el tiempo de estas dos variables.

$$\frac{dU}{dt} = Q_{sn} + Q_{li} + Q_p + Q_g - Q_{le} + Q_h + Q_e - Q_m \quad (1)$$

$$\frac{dSWE}{dt} = P_r + P_s - M_r - E \quad (2)$$

donde para la primera ecuación  $Q_{sn}$  es Radiación de onda corta neta en [ $\text{kJ m}^{-2}\text{h}^{-1}$ ],  $Q_{li}$ :corresponde a la Radiación de onda larga entrante en [ $\text{kJ m}^{-2}\text{h}^{-1}$ ],  $Q_p$  al Calor aportado por precipitación en [ $\text{kJ m}^{-2}\text{h}^{-1}$ ],  $Q_g$  es el Calor aportado por capa de sustrato bajo manto nival en [ $\text{kJ m}^{-2}\text{h}^{-1}$ ],  $Q_{le}$  la Radiación de onda larga saliente en [ $\text{kJ m}^{-2}\text{h}^{-1}$ ],  $Q_h$  es el Flujo de calor sensible en [ $\text{kJ m}^{-2}\text{h}^{-1}$ ],  $Q_e$  el Flujo de calor latente en [ $\text{kJ m}^{-2}\text{h}^{-1}$ ] y  $Q_m$  el Calor de advección eliminado por el agua de deshielo en [ $\text{kJ m}^{-2}\text{h}^{-1}$ ]. Mientras que para la ecuación (2)  $P_r$  es la tasa de precipitación líquida caída,  $P_s$  la tasa de precipitación sólida caída,  $M_r$  la tasa de agua salida de manto nival y  $E$  la sublimación/evaporación o condensación de manto nival, todas estas variables en [m/hr].

El balance energético se realiza para calcular la energía que contiene el manto nival donde solo se considera un espesor “térmicamente activo” que es donde ocurren los intercambios energéticos. En la Figura 2 se muestra un esquema representativo del modelo, donde la variable "SWE" se utiliza para rastrear la cantidad de agua presente en el manto nival, mientras que la variable "U" nos proporciona información sobre el estado físico en el que se encuentra el agua rastreada por "SWE". Este valor de energía también se utiliza para determinar la temperatura promedio de la capa de nieve y la cantidad de agua en estado líquido dentro del manto. También se considera el albedo de la nieve como una tercera variable de estado, la cual se calcula mediante parámetros relacionados con la edad de la superficie de la nieve. Para obtener más detalles, se recomienda consultar la literatura disponible sobre este modelo (Tarboton et al., 1996).

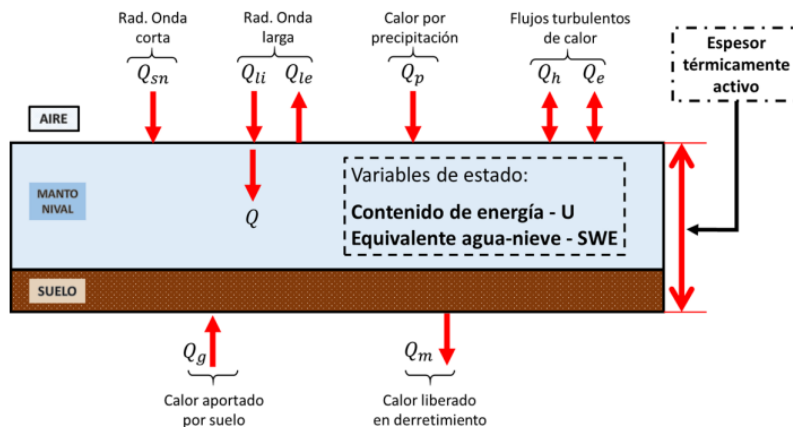


Figura 2. Esquema del modelo UEB (Sanllehi, 2023).

Este modelo ha sido modificado y aplicado en diferentes trabajos de investigación con buenos resultados en numerosas cuencas del oeste de EE.UU. (You et al., 2014), y en cuencas glaciares del Himalaya (Brown et al., 2014). A continuación, se muestra la Tabla 1 con los parámetros calibrables del modelo en este trabajo. Para más detalles sobre los rangos de los parámetros y los parámetros no calibrables del modelo se recomienda revisar el trabajo de Sanllehi (2023).

**Tabla 1.** Parámetros de variables calibrables del modelo nival UEB

Parámetro	Notación	Rango valores
Rugosidad aerodinámica de superficie [m]	$z_0$	[0.001 – 0.005]
Conductancia térmica superficial [m/hr]	$K_s$	[0.01 – 0.2]
Albedo nieve nueva espectro visible [-]	$\alpha_{v0}$	[0.8 – 0.95]
Factor de mayoración de precipitación [-]	$F_{mo}$	[1 - 3]

A continuación, se muestran las ecuaciones asociadas a las variables calibradas, las cuales forman parte de los cálculos internos del modelo de derretimiento nival UEB.

$$Ri = \frac{g (T_z - T_s)(z - z_0)}{\bar{T} U_z^2} \quad (3)$$

La ec(3) corresponde al número de Richardson que permite representar la estabilidad de la capa límite atmosférica. Donde,  $g$  corresponde a la aceleración de la gravedad ( $9.81 \text{ [m}\cdot\text{s}^{-2}]$ ), mientras que  $\bar{T}$  representa la temperatura media del aire expresada en Kelvin. Las temperaturas  $T_z$  y  $T_s$  corresponden a la temperatura del aire medida a una altura  $z$  y a la temperatura de la superficie, respectivamente y ambas en [K]. La diferencia entre estas temperaturas, junto con la altura  $z$  y la rugosidad aerodinámica  $z_0$ , determina el gradiente térmico vertical y la resistencia que presenta la superficie al intercambio turbulento. La velocidad del viento  $U_z$ , medida a la misma altura  $z$ , influye en la magnitud de la turbulencia generada; valores más altos de velocidad del viento tienden a favorecer la mezcla turbulenta y reducir la estabilidad atmosférica. Finalmente, la rugosidad aerodinámica  $z_0$  [m] describe la irregularidad de la superficie y controla la eficiencia del intercambio de momento, calor y vapor entre la superficie y la atmósfera.

$$CCIM = K_s * \rho_{nieve} * C_s * (T_s - T_m) \quad (4)$$

La ec.(4) representa la conductividad térmica de la nieve, en metros por hora, y es un parámetro que controla la transferencia de calor hacia el interior del manto nival.  $\rho_{nieve}$  es la densidad de la nieve, en  $[\text{Kg}/\text{m}^3]$ , que influye en la capacidad del manto de almacenar y transmitir calor.  $C_s$  es la capacidad calorífica específica del hielo o nieve, definida como  $2.09 \text{ [kJ/kg}\cdot\text{°C}]$ , que indica cuánta energía se requiere para cambiar la temperatura de la nieve.  $T_s$  corresponde a la temperatura superficial del manto nival en  $[\text{°C}]$ , mientras que  $T_m$  es la temperatura promedio o interior del manto nival, también en  $[\text{°C}]$ . La conducción de calor hacia el interior de la nieve  $CCIM$  se expresa en  $[\text{kJ}/\text{m}^2\cdot\text{hr}]$ .

$$\alpha_{v0} = (1 - C_v * F_{age}) \quad (5)$$

La ec.(5) representa el albedo visible de la nieve, es decir, la fracción de radiación solar reflejada por la superficie nevada en el espectro visible.  $\alpha_{v0}$  es el albedo de la nieve fresca en el espectro visible.  $C_v$  es un coeficiente que controla cómo el albedo disminuye con el envejecimiento de la nieve, mientras que  $F_{AGE}$  indica la fracción de envejecimiento de la

nieve, variando de 0 para nieve recién caída a 1 para nieve completamente envejecida. La combinación de estos términos permite que el albedo visible disminuya progresivamente a medida que la nieve envejece, afectando directamente la cantidad de radiación solar absorbida y, por tanto, la tasa de derretimiento del manto nival.

$$\text{Precipitación cuenca} = \text{Precipitación medida} * F_{mo} \quad (6)$$

Finalmente, la ec.(6) corresponde a un parámetro de calibración nuevo externo del modelo UEB que se calibra en conjunto con los demás parámetros del modelo. Este parámetro fue ingresado debido a que el modelo nival no era capaz de simular de manera aceptable los patrones nivales de las cuencas de estudio (Sanllehi; 2023). Donde los datos de precipitación se multiplican por el factor  $F_{mo}$  antes de entrar a los cálculos del modelo.

### 2.3. SUBMODELO GLACIAL GRADO ÍNDICE MEJORADO (ETI)

El modelo “Enhanced temperature-index” (ETI, nombre en idioma original) proviene del modelo estándar “grado-día”, conservando la misma estructura, pero incluye la radiación de onda corta para el cálculo del derretimiento glaciar, convirtiéndolo en un modelo nivo-glacial más bien conceptual, pero con mayor base física, permitiendo un mejor rendimiento del modelo con respecto al modelo original (Pellicciotti et al., 2005). La ecuación por la que se rige el modelo corresponde a la ec. (7).

$$M = \begin{cases} TF * T + (SRF) * (1 - \alpha_{ice}) * I & , \quad T > T_T \\ 0 & , \quad T \leq T_T \end{cases} \quad (7)$$

donde M es la Tasa de derretimiento glaciar en [mm w.e.  $h^{-1}$ ], TF corresponde al Factor de temperatura en [ $mm h^{-1} \text{ } ^\circ C^{-1}$ ], SRF al Factor de radiación de onda corta en [ $mm h^{-1} W^{-1} m^2$ ],  $\alpha_{ice}$  es el Albedo de hielo glaciar [-], I la Radiación de onda corta en [ $W m^{-2}$ ], T la Temperatura del aire y  $T_T$  la Temperatura umbral donde hay derretimiento, ambas temperaturas en [ $^\circ C$ ].

La naturaleza conceptual del modelo restringe su aplicabilidad a glaciares específicos que cumplan con el requisito de ser dominados por procesos de derretimiento por fusión en lugar de evapotranspiración y/o sublimación (Ayala et al., 2020). Sin embargo, el modelo ETI ha sido implementado y validado en varias ocasiones anteriores en la zona central de Chile, más específicamente en cuencas de la cordillera de los Andes (Ayala et al., 2020; Burger et al., 2019; Shaw et al., 2020), mostrando mejor rendimiento en glaciares donde su balance energético está dominado por la radiación de onda corta y que presentan temperaturas medias elevadas, y encontrándose en un rango de cotas de 3000 a 4500 [msnm]. Para calibraciones anteriores de este modelo se han utilizado los parámetros descritos en la Tabla 2, los cuales se deben obtener para cada glaciar presente en la cuenca. Al igual que para el modelo anterior, se recomienda revisar el trabajo de Sanllehi (2023) para más detalles relacionados a los parámetros que utiliza el submodelo

**Tabla 2.** Parámetros de variables caliobrables del modelo glaciar ETI

Parámetro	Notación	Rango valores
Factor de temperatura [ $mm h^{-1} \text{ } ^\circ C^{-1}$ ]	$TF$	[0 – 0.12]
Factor de radiación de onda corta [ $mm h^{-1} W^{-1} m^2$ ]	$SRF$	[0 – 0.01078]
Temperatura umbral donde hay derretimiento [ $^\circ C$ ]	$T_T$	[-2 – 6]

## 2.4. SUBMODELO DE ESCORRENTÍA DE MIGUELES

Este modelo se encarga de recibir el derretimiento glacial y nivel de los modelos descritos anteriormente, simulando un balance de humedad del suelo. Para esto el modelo consta de una capa superficial y otra subterránea, considerando la infiltración de la capa superior al subsuelo y la evapotranspiración para generar un caudal de escurrimiento superficial y otro subsuperficial. Este modelo es de carácter semi distribuido y conceptual, ya que, considera un almacenamiento tipo estanque con cálculos que permiten una resolución diaria (Migueles, 1994). El modelo de escorrentía de Migueles ha sido utilizado en numerosos trabajos anteriores para la generación de caudales hidrológicos (Flores, 2017; Vargas, 2022). A modo de ejemplo de cómo funciona el modelo, se puede visualizar un esquema en la Figura 3 donde se aplican las ecuaciones principales de este submodelo, como el balance de humedad del suelo que se representa mediante el cálculo de humedad en un embalse superficial con una capacidad máxima definida como **HSMAX**. Este “embalse” se recarga con el agua que se infiltra en el suelo considerando pérdidas de humedad por efecto de la evaporación. Donde en la ec. (8),  $HS(i, j)$  es la Humedad en el suelo en la banda  $i$  en el día  $j$ , [m],  $D(i, j)$  el Agua entrante al modelo en la banda  $i$  en el día  $j$ , [m], y  $EVAP(j)$  la Evapotranspiración total en la banda  $i$  en el día  $j$ , [m]. Siendo  $FRAC$  un parámetro calibrable del modelo que representa el porcentaje de agua que se infiltra en el suelo. Mientras que para la ec. (9)  $DI$  es el agua excedente por sobre  $HSMAX$  que escurre subterráneamente, [m]. En la Tabla 3 se pueden apreciar todos los parámetros que se dejaron como “calibrables” para este trabajo. Para más detalles revisar el trabajo de Sanllehi (2023).

$$HS(i, j) = HS(i, j - 1) + FRAC \cdot D(i, j) - EVAP(j) \quad (8)$$

$$\begin{aligned} DI(i, j) &= HS(i, j) - HSMAX && ; HS(i, j) > HSMAX \\ DI(i, j) &= 0 && ; HS(i, j) \leq HSMAX \end{aligned} \quad (9)$$

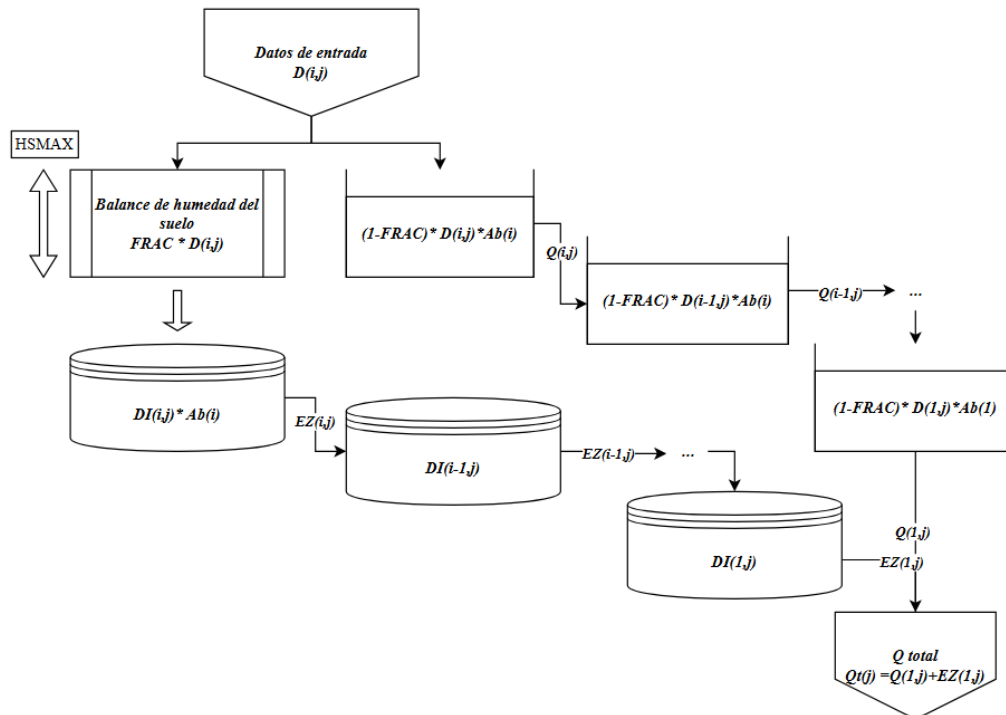


Figura 3. Esquema modelo de síntesis de escorrentía.

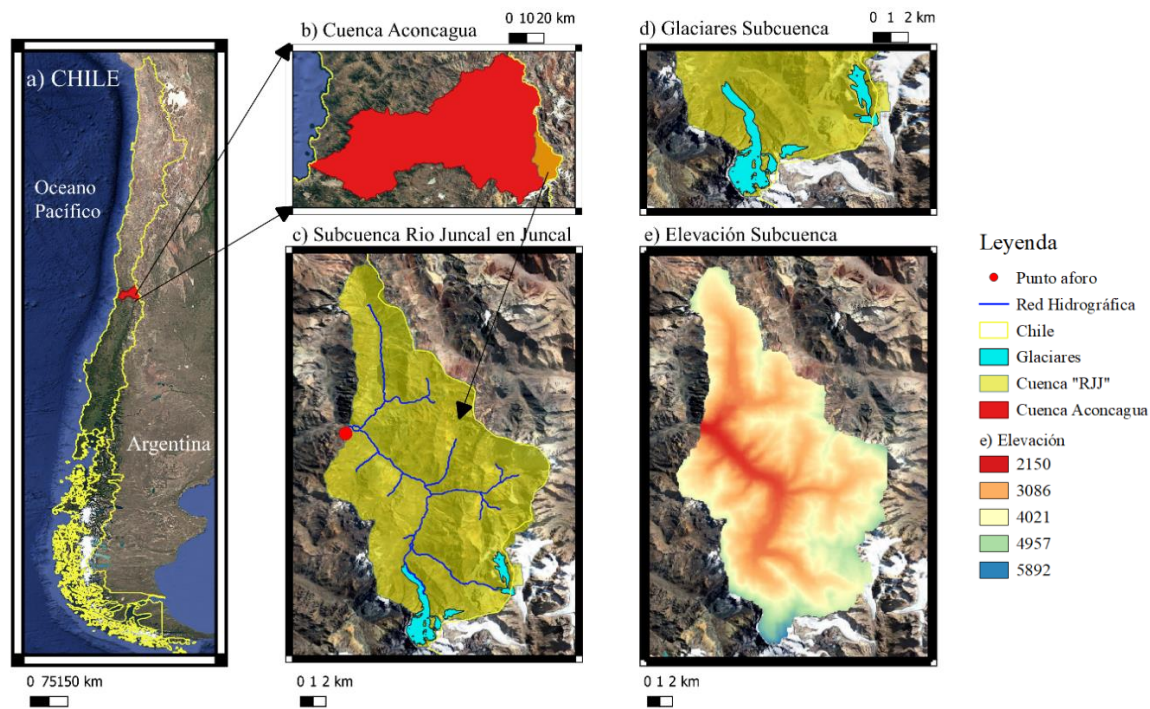
**Tabla 3.** Parámetros de variables caliobrables del modelo de escorrentia

Parámetro	Notación	Rango valores
Fracción de agua que infiltra en el suelo [-]	FRAC	[0 - 1]
Capacidad máxima de almacenamiento de agua del suelo [m]	HSMAX	[0 - 50]
Coefficiente de descarga de embalse lineal subterráneo [1/día]	CK	[0.0001 - 0.1]
Constante que representa la medida del desfase de la onda decrecida [horas]	Km	[24 - 72]
Factor adimensional de ponderación de importancia de caudales de entrada y salida del volumen de control [-]	xm	[0.1 - 0.5]

## 4. METODOLOGIA

### 4.1. ZONA DE ESTUDIO

El área de estudio seleccionada corresponde a la cuenca "Río Juncal en Juncal", definida como "RJJ", a modo de simplificación. En la Figura 4 se puede observar que esta cuenca se encuentra en la región de Valparaíso, zona central de Chile. La cuenca cuenta con una elevación media de 3662 [msnm], un mínimo de 2273 [msnm] y un máximo de 5919 [msnm], abarcando un área de 343.1 [km<sup>2</sup>]. Para la discretización de la cuenca requerida por los modelos distribuido, se utilizaron pixeles de 500 [m<sup>2</sup>] resultando en un total de 1324 puntos de grilla. La elección se basa en la disponibilidad de un gran número de mediciones de caudal que se utilizaron para calibrar y validar los modelos anteriormente descritos en trabajos de investigación previos a este, correspondiendo a un régimen hidrológico de tipo nivo-glacial. Más detalladamente, se consideran 20 años de datos de series diarias de caudal (del año 2000 hasta 2019), medidos por la estación "Río Juncal en Juncal" ubicada en el punto de aforo de la cuenca que se muestra en la Figura 4.c. Para la definición de los periodos de calibración y validación, primero se ordenó en años hidrológicos, clasificando cada uno como húmedo o seco, y posteriormente se combinaron para establecer los periodos de calibración y validación. Los años finalmente seleccionados para cada etapa se presentan en la Tabla 4. Las series de caudales se utilizaron para la calibración del modelo de escorrentía, mientras que para el modelo nival UEB se consideran la series de SCA obtenidas de imágenes MODIS. En la cuenca existen 5 grandes glaciares que son considerados para la modelación: Monos de Agua, Juncal Norte (A y B) y otros dos sin ser nombrados observados en la Figura 4.d. Para la calibración del Modelo glaciar ETI se considera la tasa de variación de elevación glaciar medio entre 2000-2018 presentadas como resultados en el trabajo de Dussailant et al. (2019).



**Figura 4.** Mapa con ubicación geográfica de cuenca en estudio.

**Tabla 4:** Periodos de calibración y validación utilizados en métricas de evaluación de los modelos.

Calibrados	2002, 2003, 2004, 2006, 2009, 2011, 2012, 2014, 2017, 2018
Validados	2000, 2001, 2005, 2007, 2008, 2010, 2013, 2015, 2016, 2019

Los modelos también requieren datos de precipitación, temperatura máxima y temperatura mínima como variables de entrada. Estos datos son extraídos del producto grillado diario disponible por CR2 para la misma ventana de tiempo antes mencionada. Cabe mencionar que para la calibración del modelo Sanllehi (2023) utiliza la versión v2.0 de este producto con correcciones pertinentes descritas en su trabajo. Sin embargo, para la nueva calibración implementada en Python se utiliza la versión v2.5 del mismo producto.

## 4.2. IMPLEMENTACIÓN

Para implementar el modelo en Python se realizaron 2 tareas principales, primero se llevó a cabo la traducción del código fuente del modelo glacio-hidrológico USM que consistió en realizar un análisis detallado del modelo para comprender su estructura modular, flujos de información y algoritmos de cálculo. Se generaron esquemas para visualizar la estructura interna del modelo y comprender como se conectan las funciones encargadas de ejecutar los algoritmos de cálculo de cada submodelo. Posteriormente, se realizó su traducción funcional al lenguaje Python línea por línea, asegurando la equivalencia operativa del código. En segundo lugar, se verificó el funcionamiento del nuevo modelo, donde se replicaron los resultados obtenidos por las calibraciones originales en la cuenca de análisis, comparando los resultados de las series simuladas en ambos lenguajes. Para la replicación de los resultados se utilizaron los parámetros obtenidos de las calibraciones realizadas por Sanllehi (2023), ejecutando el modelo en Python con los mismos valores numéricos para estos parámetros en cada submodelo correspondiente. También se emplearon calibraciones nuevas en Python para demostrar su aplicabilidad y versatilidad, obteniendo nuevos parámetros de

calibración para la misma zona de estudio. Finalmente, se buscó optimizar el modelo en Python con técnicas de vectorización de código y asegurando mantener los resultados logrados por la equivalencia operativa anteriormente mencionada. Se lleva a cabo una comparación de rendimiento en función del tiempo que demora en ejecutarse el modelo en cada lenguaje y en cada versión para el caso del código en Python.

Para las nuevas calibraciones, se ejecutaron todos los módulos de manera simultánea, dando énfasis a las series de caudales y calibrando respecto a series de caudales observados por una estación en el punto de aforo de la cuenca. Se obtuvieron nuevos parámetros para los submodelos de nieve UEB y de escorrentía de Migueles (1994). De esta forma los parámetros calibrados corresponden a los que se muestran en la Tabla 1 y Tabla 3. Debido a la limitación de costo computacional, los parámetros del submodelo glaciar ETI se utilizaron de manera fija y corresponden a los encontrados por las calibraciones de Sanllehi (2023), los cuales también están dentro de los rangos presentados en la Tabla 2. Respecto a los periodos para calibrar y validar el modelo, estos consisten en una combinación de los años húmedos y años secos que se presentaron en la cuenca a lo largo del periodo de estudio, y corresponden a los mismos que se utilizaron en el trabajo de Sanllehi (2023). También se implementó el algoritmo de búsqueda automática global SCE (Shuffle Complex Evolution; Duan et al., 1993), orientado a optimizar el valor de la función objetivo, que en este caso corresponde a la métrica de eficiencia Kling-Gupta (KGE). Cabe mencionar que solo las series simuladas por los submodelos UEB y de escorrentía pueden ser evaluadas mediante la métrica KGE, debido a la naturaleza de los datos observados disponibles con los cuales se realiza la comparación. Para evaluar el desempeño del submodelo ETI se utilizó el coeficiente de determinación ( $R^2$ ), siguiendo el enfoque de calibraciones anteriores realizadas por Sanllehi (2023), en las cuales también se empleó esta métrica como función objetivo del algoritmo SCE. Sin embargo, en esta nueva calibración se busca únicamente optimizar el valor de KGE obtenido a partir de las series de caudales simuladas por el modelo de Migueles (1994), para el cual la cuenca fue discretizada en 12 bandas de elevación de área relativamente equivalente, cubriendo toda la cuenca (aproximadamente 29 [km<sup>2</sup>] por banda, lo que corresponde al 8.3% del área total).

## **5. RESULTADOS**

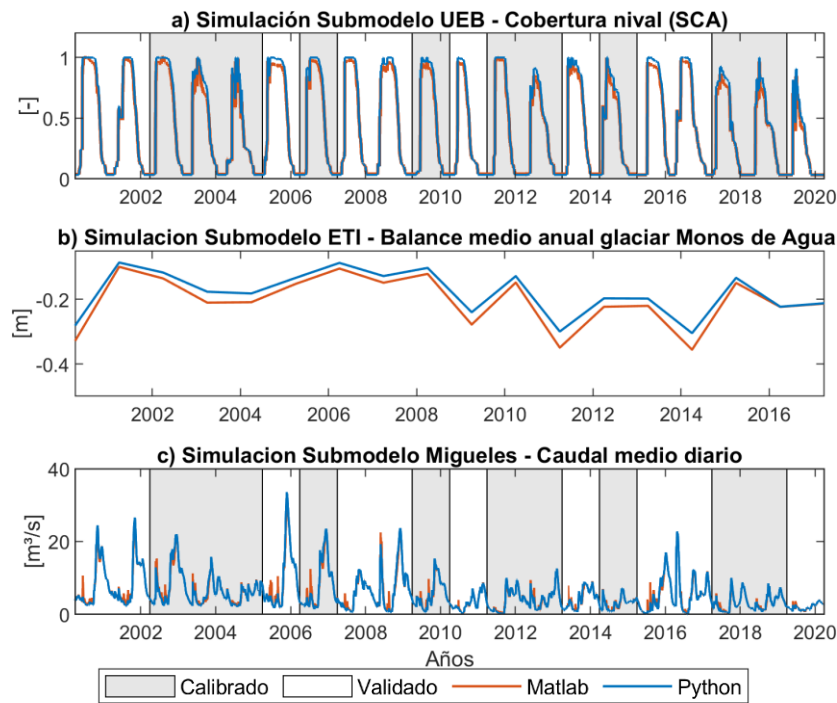
### **5.1. REPLICACIÓN**

En el caso del modelo de simulación de derretimiento de nieve de Tarboton (UEB; Figura 5a), se puede apreciar una diferencia numérica entre las series de tiempo simuladas para la variable de SCA al comparar las implementaciones en Matlab y Python. No obstante, ambas series presentan un comportamiento temporal muy similar. Estas diferencias numéricas se reflejan en los valores de la métrica KGE: 0.92 en Matlab y 0.88 en Python durante el periodo de calibración; y 0.90 en Matlab frente a 0.86 en Python en el periodo de validación. Estas discrepancias se atribuyen a las diferencias en los métodos numéricos incorporados por defecto en cada lenguaje, las cuales se acumulan a lo largo de las múltiples ecuaciones del modelo nivales UEB.

Para el submodelo glaciar (ETI), se compararon las simulaciones del glaciar Monos de Agua entre Python y Matlab (Figura 5b). Aunque se simularon varios glaciares, se presenta este caso representativo debido a que el comportamiento fue similar en todos. Las series

corresponden al balance másico medio anual simulado. Al igual que con el modelo UEB, se identifican diferencias numéricas entre las series, pero su evolución temporal es coherente. La discrepancia máxima entre ambas series alcanza los 0.05 [m]. Esta diferencia también se evidencia en los valores del coeficiente de determinación ( $R^2$ ), con 0.44 en Matlab y 0.77 en Python, considerando todos los glaciares simulados.

Finalmente, para el modelo de escorrentía, se compararon las series de caudal medio diario simuladas por el submodelo de Migueles (1994) en Matlab y su réplica en Python (Figura 5c). Si bien se identifican diferencias numéricas a lo largo de toda la serie, los patrones generales y tendencias se mantienen. Las discrepancias en los valores promedios son pequeñas ( $0.02 \text{ m}^3/\text{s}$  aproximadamente), aunque algunos peaks muestran diferencias de hasta  $6 \text{ [m}^3/\text{s}]$ , mientras que las diferencias más pequeñas son del orden de  $10^{-16}$ , sin alcanzar coincidencia exacta. En cuanto al desempeño, ambos modelos alcanzan un valor de KGE de 0.67 en el periodo de calibración. En validación, el valor fue de 0.57 en Python y 0.59 en Matlab, con una diferencia de 0.02.



**Figura 5.** Resultados simulaciones en Matlab y Python con el mismo set de parametros.

## 5.2. RENDIMIENTO

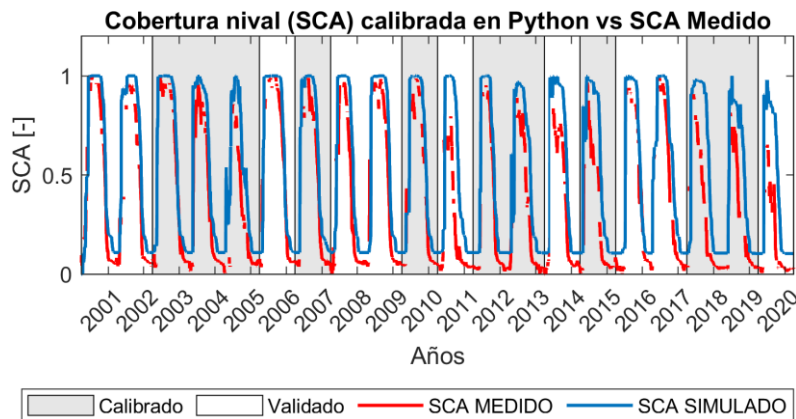
A modo de comparar el rendimiento del modelo en los diferentes lenguajes de programación, se registró el tiempo de cómputo de una simulación completa y se estimó el tiempo promedio por punto de grilla, ya que, el modelo nival UEB concentra la mayor carga de cómputo y es de carácter distribuido, al igual que el modelo glaciar ETI; sin embargo, como los glaciares ocupan un área significativamente menor respecto al total de la cuenca, se representan con menos puntos. Primero se midió el tiempo en Matlab, luego en Python para la versión inicial de la traducción y, finalmente, para una versión optimizada en Python. Todas las mediciones se realizaron en el mismo equipo de cómputo; los valores absolutos pueden variar según el hardware, pero las relaciones relativas entre implementaciones deberían mantenerse aproximadamente.

**Tabla 5:** Tiempo aproximado por punto de grilla que demora el programa en ejecutar una simulación.

Matlab	Python sin optimizar	Python optimizado
8.15 [s]	16.5 [s]	1 [s]

### 5.3. APLICACIÓN

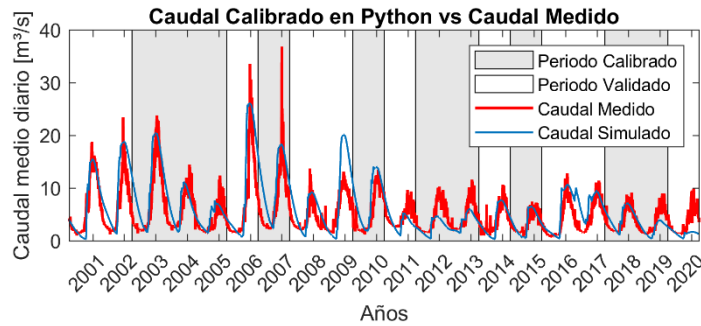
A continuación, se muestran los resultados de una nueva calibración realizada en Python, la cual entregó un conjunto actualizado de parámetros para los submodelos UEB y de escorrentía de Migueles (1994) en la cuenca “RJJ”. En primer lugar, la Figura 6 muestra simulación de SCA generada por el submodelo de derretimiento nival UEB utilizando los parámetros obtenidos en dicha calibración. Cabe señalar que la calibración no se realizó utilizando SCA como variable objetivo, pero se evaluó igualmente su desempeño debido a la relevancia de este proceso dentro del ciclo de derretimiento en la cuenca. Si bien la simulación representa adecuadamente el comportamiento general de los datos medidos, no logra replicar con precisión sus valores numéricos. Esto se refleja en los valores de la métrica KGE: 0.51 para el periodo calibración y 0.49 para el de validación, ambos inferiores a los obtenidos en la calibración realizada en Matlab.



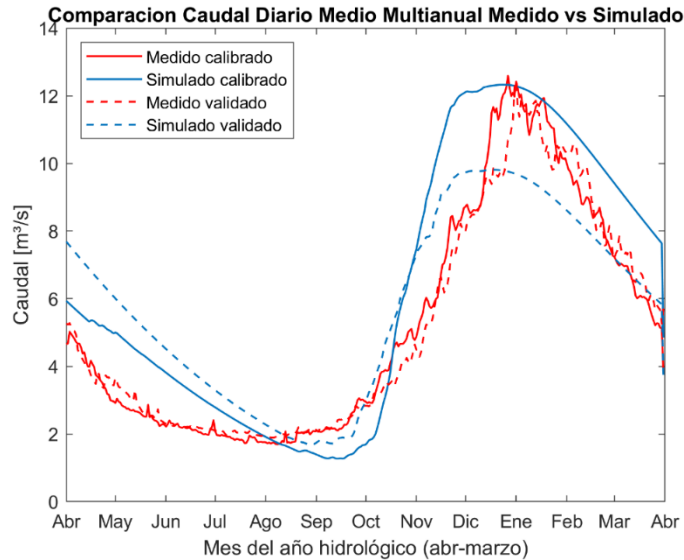
**Figura 6.** Serie de SCA simulada por el submodelo UEB con parámetros nuevos vs. datos observados.

En segundo lugar, la Figura 7 presenta los resultados de la simulación del submodelo de escorrentía con los nuevos parámetros obtenidos mediante la calibración en Python. Se alcanzaron mejores valores de la métrica KGE en comparación con la calibración realizada en Matlab, con un valor de 0.75 para el periodo de calibración y 0.67 para el de validación. Aunque el ajuste general mejora respecto a datos observados, se aprecia que en varias zonas la simulación subestima el caudal, especialmente en los peaks observados, y presenta valores más bajos en los valles.

La Figura 8, que presenta las series de caudales medios multianuales para los periodos de calibración y validación, permite observar un desfase en los peaks de escorrentía. En particular, se aprecia que la escorrentía máxima ocurre anticipadamente por cerca de un mes y medio, lo que implica una sobreestimación del caudal en etapas tempranas en el periodo de calibración y una subestimación en el peak de escorrentía durante el periodo de validación. Además, la simulación muestra un comportamiento más regular, sin cambios bruscos en el caudal, lo que sugiere posibles limitaciones del modelo para representar adecuadamente la estacionalidad en la cuenca.



**Figura 7.** Caudal resultante de la nueva calibración del modelo en Python vs. datos medidos.



**Figura 8.** Caudal Multianual de la nueva calibración en Python.

## 6. ANALISIS Y CONCLUSIONES

En cuanto a la replicación de los submodelos, se logró reproducir el comportamiento esperado en todos los casos. No obstante, se observaron diferencias numéricas entre los resultados obtenidos en Matlab y Python, a pesar de la traducción detallada del código línea por línea. Estas discrepancias pueden atribuirse al arrastre de errores numéricos inherentes a cada entorno de programación, especialmente por las diferencias en la gestión de la precisión y redondeo en operaciones matemáticas sucesivas, propias de los modelos hidrológicos. Sin embargo, estas diferencias resultaron poco significativas en el contexto de la calibración, ya que la obtención de nuevos parámetros ajustados al entorno de Python permitió alcanzar desempeños similares o incluso superiores respecto a Matlab. Así lo demuestran los resultados de la recalibración, que mejoran el ajuste de los caudales simulados en la cuenca.

A pesar de esta mejora, era esperable que al replicar los resultados del submodelo UEB el desempeño se viera afectado negativamente, debido a que al calibrar con base en caudales no se supervisa explícitamente la correcta representación de los procesos físicos del submodelo nival. Esto ocurre porque todos los submodelos se ejecutan antes de calcular el valor entregado por la función objetivo (KGE para caudales, en este caso), operando como una especie de “caja negra”. En este contexto, cobra especial relevancia la “calibración por etapas” que presenta Sanllehi (2023), si se busca representar de forma más fiel los distintos procesos hidrológicos presentes en la cuenca.

Respecto a la calibración del submodelo glaciar ETI, cabe señalar que la limitada disponibilidad de datos reduce la robustez del proceso. Calibrar un modelo de resolución horaria a partir de un único valor observado anual (el balance de masa glaciar medio anual) disminuye significativamente el valor añadido de trabajar con un enfoque distribuido y de alta resolución temporal. Esta limitación responde exclusivamente a la escasez de datos disponibles, por lo que la incorporación futura de series horarias observadas permitiría aprovechar plenamente el potencial del modelo, posibilitando una calibración más detallada y representativa.

El costo computacional representó también una limitación importante. Durante la calibración en Python fue necesario restringir el número de iteraciones y reducir la resolución espacial con respecto a lo utilizado en Matlab, con objetivo de reducir los tiempos de ejecución. Si bien aumentar el número de iteraciones permitiría una búsqueda más exhaustiva de parámetros óptimos, esto implicaría tiempos de cómputo significativamente mayores. Por lo tanto, se identifica este aspecto como un área clave para futuras mejoras. Cabe mencionar que la calibración del modelo en Python se realizó antes de optimizar el código por temas de prioridad. Sin embargo, los resultados de rendimiento obtenidos al optimizar el código muestran una mejora significativa, permitiendo que el modelo pueda ser calibrado de una forma más robusta en trabajos futuros.

En síntesis, la replicación del modelo en lenguaje Python puede considerarse exitosa, logrando simular caudales en el punto de aforo de la cuenca con resultados de calibración satisfactorios. Si bien este trabajo no presenta grandes novedades desde el punto de vista hidrológico, permite reforzar los comportamientos observados en el régimen nivo-glaciar de la cuenca descritos en trabajos anteriores, sirviendo además como un punto de partida para futuras mejoras y acoplamientos del modelo con módulos más robustos que permitan simular la escorrentía de manera completamente distribuida. El hecho de haber implementado el modelo en Python, un lenguaje de código abierto, entrega una alta flexibilidad para modificar su estructura e incorporar nuevas funciones o submodelos hidrológicos en el futuro, potenciando así su aplicabilidad y escalabilidad.

No obstante, se identificaron oportunidades de mejora importantes, especialmente en términos de eficiencia computacional. En particular, la simulación de un punto de grilla resultó aproximadamente dos veces más rápida en Matlab que en Python sin aplicar optimizaciones. Sin embargo, al tratarse de un lenguaje abierto y altamente adaptable, Python permite optimizar el rendimiento de forma considerable. Esto se comprobó al vectorizar partes del código, logrando una reducción significativa en los tiempos de simulación y evidenciando el potencial del modelo para seguir mejorando su desempeño y escalabilidad en futuras versiones. Una alternativa recomendada a trabajos futuros es aplicar técnicas de paralelización para aprovechar al máximo la capacidad computacional disponible. Otra opción sería adaptar el código para su ejecución en unidades de procesamiento gráfico (GPU), aprovechando tecnologías actualmente disponibles que permiten un uso más eficiente de estos recursos, tal como lo demuestra SWEpy (Fuenzalida et al.; 2025), un modelo de inundación en Python que resuelve ecuaciones complejas y con el cual el modelo glacio-hidrológico USM presentado en este trabajo podría acoplarse.

## AGRADECIMIENTOS

Esta investigación fue financiada por el proyecto ANID FONDECYT de Iniciación N.º 11250238 y por el proyecto interno de investigación PI\_LIR\_24\_01 de la Universidad Técnica Federico Santa María.

## REFERENCIAS

- Ayala, Á., Fariás-Barahona, D., Huss, M., Pellicciotti, F., McPhee, J., and Farinotti, D.: Glacier runoff variations since 1955 in the Maipo River basin, in the semiarid Andes of central Chile, *The Cryosphere*, 14, 2005–2027, <https://doi.org/10.5194/tc-14-2005-2020>, 2020.
- Brown, M., Racoviteanu, A., Tarboton, D., Gupta, A. S., Nigro, J., Policelli, F., . . . Tokar, S. (2014). An integrated modeling system for estimating glacier and snow melt driven streamflow from remote sensing and earth system data products in the himalayas. *Journal of Hydrology*, 519 (18), 1859–1869. doi: 10.1016/j.jhydrol.2014.09.050.
- Burger, F., Ayala, A., Fariás, D., Shaw, T. E., MacDonell, S., Brock, B., . . . Pellicciotti, F. (2019). Interannual variability in glacier contribution to runoff from a high-elevation andean catchment: understanding the role of debris cover in glacier hydrology. *Hydrological Processes*, 33 (2), 214–229. doi: 10.1002/hyp.13354.
- CR2. (2015). Informe a la nación. la megasequia 2010-2015: Una lección para el futuro. En (p. 26). Santiago, Chile: Informe a la Nación.
- Duan, Q. Y., Gupta, V. K., y Sorooshian, S. (1993, mar). Shuffled complex evolution approach for effective and efficient global minimization. *Journal of Optimization Theory and Applications*, 76(3), 501–521. doi: 10.1007/BF00939380
- Dussaillant, I., Berthier, E., Brun, F., Masiokas, M., Hugonnet, R., Favier, V., ... Ruiz, L. (2019). Two decades of glacier mass loss along the andes. *Nature Geoscience*, 12(10), 802–808. doi: 10.1038/s41561-019-0432-5
- Fuenzalida, J., Kusanovic, D., Meza, J., Meneses, R., & Catalan, P. (2025). SWEpy: An Open-Source GPU-Accelerated Solver for Near-Field Inundation and Far-Field Tsunami Modeling. *EGU sphere*, 2025, 1-46.
- Flores, D. (2017). Metodologías para predicción de caudales medios diarios utilizando un modelo hidrológico de derretimiento de nieve, y transformación de derretimiento en escorrentía aplicado en la cuenca del río juncal. Valparaíso, Chile: Universidad Técnica Federico Santa María.
- Garreaud, R. D., Boisier, J. P., Rondanelli, R., Montecinos, A., Sepúlveda, H. H., & Veloso-Aguila, D. (2020). The Central Chile Mega Drought (2010–2018): A climate dynamics perspective. *International Journal of Climatology*, 40(1), 421–439. <https://doi.org/10.1002/joc.6219>
- Miguelés, R. (1994). Síntesis de Caudales Diarios en Cuencas Pluvionivales. Valparaíso: Universidad Técnica Federico Santa María
- Pellicciotti, F., Brock, B., Strasser, U., Burlando, P., Funk, M., y Corripio, J. (2005). An enhanced temperature-index glacier melt model including the shortwave radiation balance: Development and testing for haut glacier d'arolla, switzerland. *Journal of Glaciology*, 51 (175), 573–587. doi: 10.3189/172756505781829124
- Sanllehi, A. (2023). Integración y evaluación de modelo de derretimiento y acumulación de nieve y hielo glaciario a modelo hidrológico USM. Valparaíso, Chile: Universidad Técnica Federico Santa María.
- Shaw, T., Caro, A., Mendoza, P., Ayala, A., Pellicciotti, F., Gascoin, S., y McPhee, J. (2020). The utility of optical satellite winter snow depths for initializing a glacio-hydrological model of a high-elevation, andean catchment. *Water Resources Research*, 56 (8). doi: 10.1029/2020WR027188
- Stowhas, L. (1975). Simulation of Energy Budget and Melting Processes in Snowpack Systems under Data Constrains. Davis: University of California
- Tarboton, D., y Luce, C. (1996). Utah energy balance snow accumulation and melt model (ueb). computer model technical description and users guide (Inf. Téc.).
- Van Tiel, M., Stahl, K., Freudiger, D., y Seibert, J. (2020). Glacio-hydrological model calibration and evaluation. *Wiley Interdisciplinary Reviews: Water*, 7 (6). doi: 10.1002/wat2.1483
- Vargas, G. (2022). Simulación de caudales diarios en cuencas andinas del centro de Chile empleando datos meteorológicos en cr2met en un modelo hidrológico semi-distribuido. Valparaíso, Chile: Universidad Técnica Federico Santa María.
- You, J., Luce, C., y Tarboton, D. (2014). Modeling the snow surface temperature with a onelayer energy balance snowmelt model. *Hydrology and Earth System Sciences*, 14 (18), 5061–5076. doi: 10.5194/hess-18-5061-2014.