

# Universidad Autónoma de Baja California

---

## Facultad de Ingeniería, Arquitectura y Diseño

### Paradigmas de la programación

#### Práctica 3

#### *Instalación y funcionamiento de Haskell*

Arturo Rafael Cornejo Escobar

31 de abril del 2025

---

## INTRODUCCIÓN

El objetivo de esta práctica es identificar los **elementos esenciales de los lenguajes de programación**, incluyendo: **nombres**, **marcos de activación**, **bloques de alcance**, **administración de memoria**, **expresiones**, **comandos**, **control de secuencia** (selección, iteración y recursión), **subprogramas** y **tipos de datos** en el ámbito de Python a diferencia de la práctica pasada que era de C.

Por lo que es importante marcar las diferencias en los elementos:

Los **nombres** manejan un tipado dinámico, lo que se entiende como qué el tipo de una variable se es determinado durante el tiempo de la ejecución del programa y no durante la compilación, lo que abre la posibilidad de cambiar el tipo según el dato agregado.

En el caso de los **marcos de activación**, son manejados de forma automática también, esto infiere que cuando se llama a una función o sub programa, Python crea el entorno en la pila para las variables locales y las referencias, además de limpiarlo al finalizar, mientras que en C es un manejo más manual sobre la gestión de la pila.

La primera diferencia que se puede notar, son los **bloques de alcance** ya que, están delimitados por la indentación a diferencia de definirlo por llaves ("{"}) y los **comandos** se terminan con una nueva línea y agrupados según la indentación.

---

## DESARROLLO

En la página oficial de Haskell, nos encontramos con el comando del [instalador](#).

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; try { &  
([ScriptBlock]::Create((Invoke-WebRequest  
https://www.haskell.org/ghcup/sh/bootstrap-haskell.ps1 -  
UseBasicParsing))) -Interactive -DisableCurl } catch { Write-Error $_ }
```

---

Después de confirmar los directorios de instalación y la instalación de módulos, se abre una ventana de MSY64 que terminará la descarga de los elementos necesarios.

Una vez terminada la instalación, hacemos verificación de esta, con el comando `gch --version` y regresa:

```
The Glorious Glasgow Haskell Compilation System, version 9.6.7
```

Se procede a instalar [Docker](#) que es una herramienta que te permite **crear, ejecutar y compartir aplicaciones en contenedores**.