

Introducción al entorno de programación Racket

El entorno de programación que se va a utilizar para *Scheme* es *DrRacket*. Este entorno está desarrollado por la Universidad Rice de Texas bajo licencia GNU y se puede descargar desde su sitio Web (<http://racket-lang.org/>). Este entorno se desarrolló con un objetivo educacional, lo que se refleja en el uso de distintos niveles de lenguaje *Scheme*. En nuestro caso particular trabajaremos habitualmente con el correspondiente al estándar *Revised5 Report on the Algorithmic Language Scheme (Legacy Languages/R5RS)* o su versión con muchas más funciones (*Legacy Languages/Pretty Big*).

En los siguientes apartados se configura y describe el IDE para *Scheme* así como las tareas típicas de edición, depuración y evaluación.

Configuración del IDE

La primera vez que se entra al IDE es necesario especificar el lenguaje que se va a utilizar. Hasta que no se especifique este no se puede ejecutar un programa, tal y como se indica en la ventana inferior del entorno. Para elegir el lenguaje hay que abrir el menú desplegable *No language chosen* y seleccionar la opción *Choose Language...*, tal y como se muestra en Figura 1 y Figura 2.

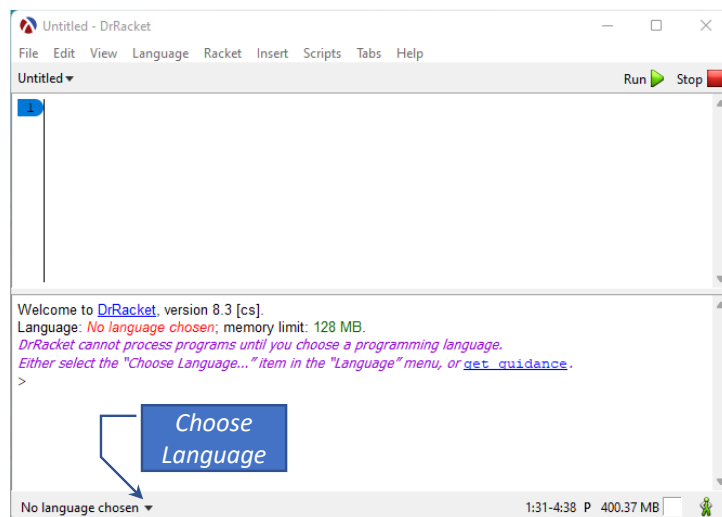


Figura 1. Primera vista del entorno

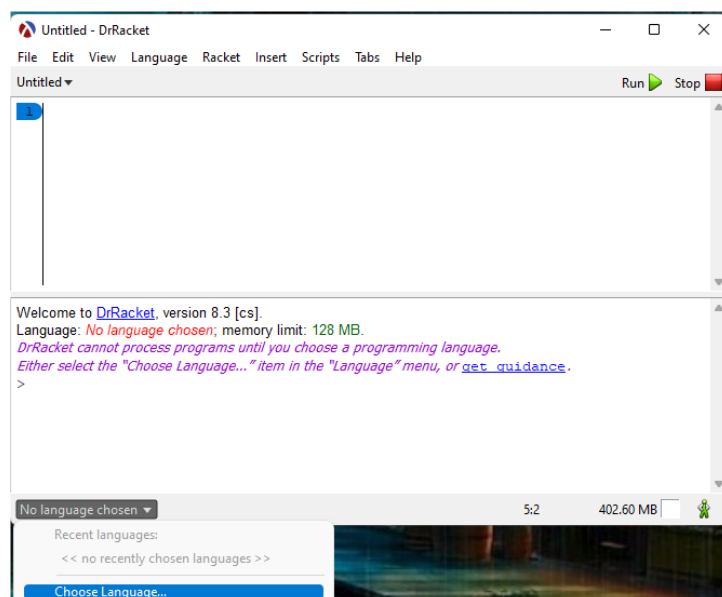


Figura 2. Elijiendo la versión del lenguaje

En el cuadro de diálogo que se muestra hay que marcar la casilla *Other Languages* y seleccionar el nivel de lenguaje *Pretty Big* (ver Figura 3).

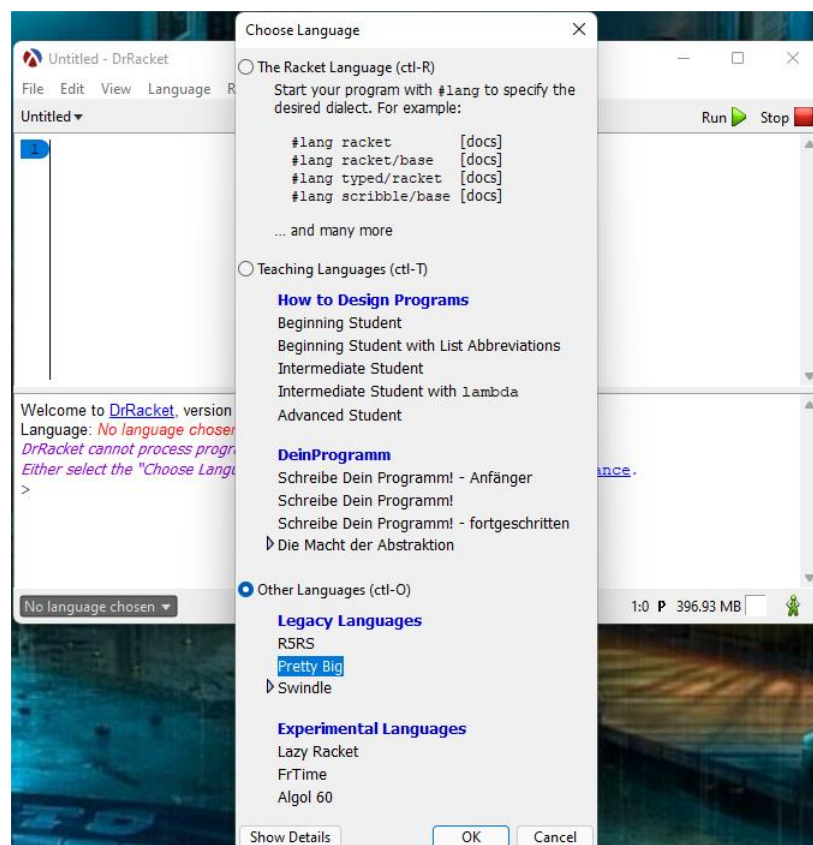


Figura 3. Selección del lenguaje *Pretty Big*

Una vez que pulsemos el botón OK habremos seleccionado el lenguaje a utilizar, el entorno queda habilitado para ejecutar programas y, en lo sucesivo, la ventana que se mostrará al abrir éste tendrá un aspecto similar al que se muestra en la Figura 4

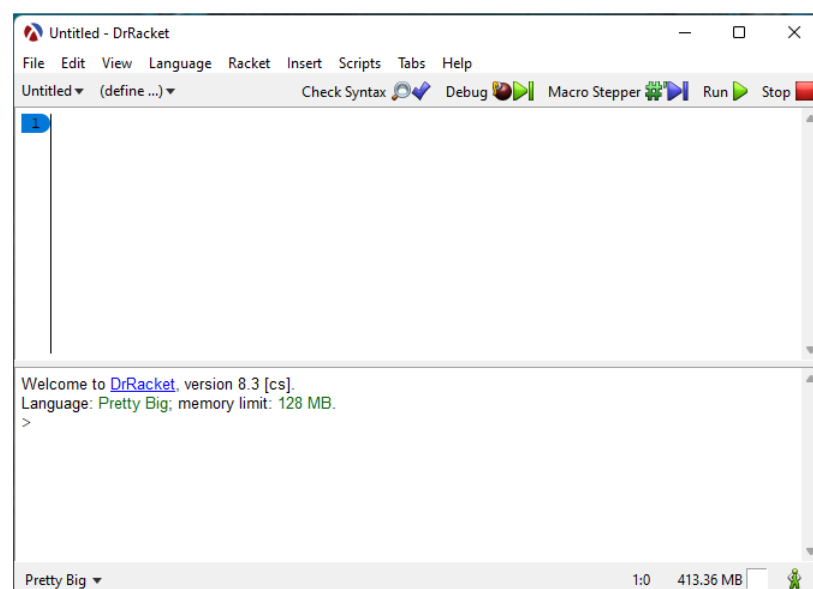


Figura 4. DrRacket con lenguaje *Pretty Big*

Otras opciones de configuración

Con excepción de la selección del lenguaje a utilizar, todas las opciones de configuración del IDE se realizan en el cuadro de diálogo que se muestra al desplegar el menú *Edit* y seleccionar su última opción: *Preferences...* (ver Figura 5).

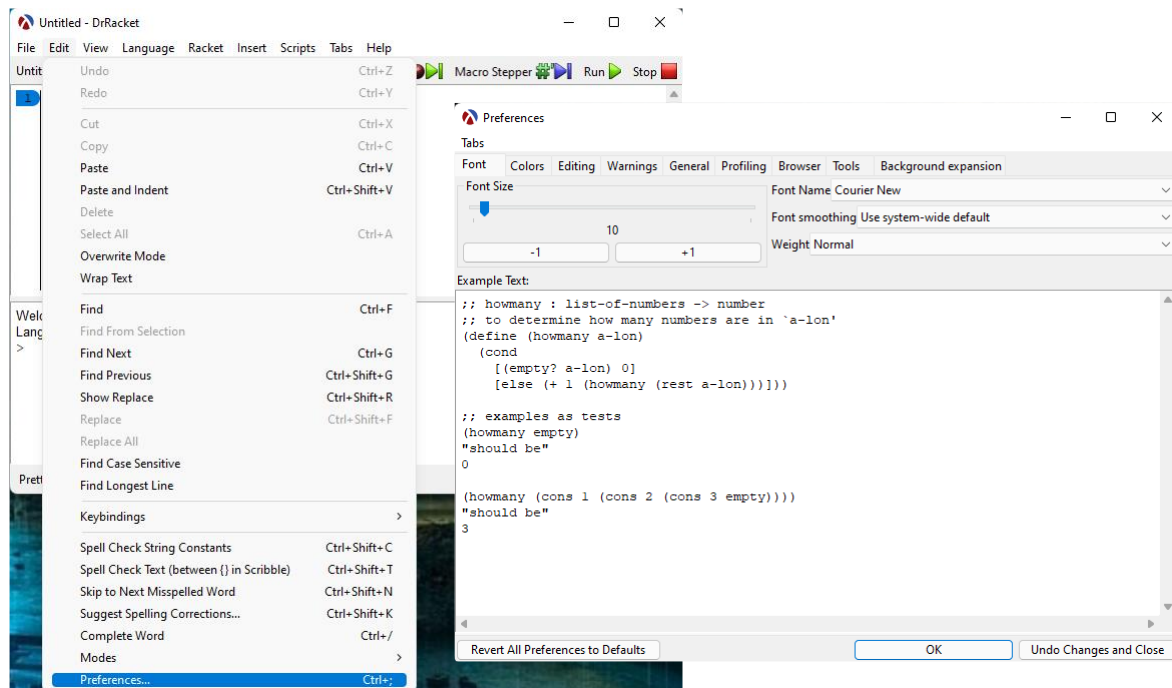


Figura 5. Cuadro de diálogo de preferencias

Al margen de otras preferencias que se quieran establecer, se recomienda incrementar el tamaño de la fuente como mínimo al 12.

Descripción de la interfaz

Una vez cargado el entorno (DrRacket.exe) se nos presenta la interfaz de programación, similar a la que muestra la Figura 6.

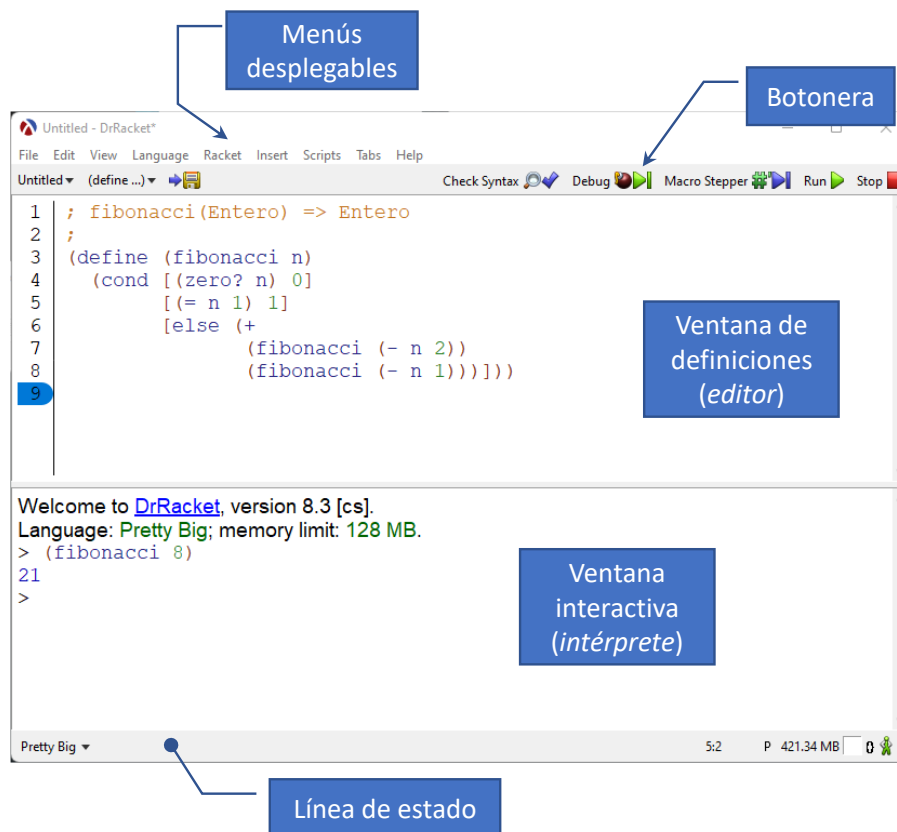


Figura 6. Interfaz gráfica de Racket

Menús

- File:

New/New Tab: Abre una nueva ventana/pestaña del entorno.

Abrir: abre un archivo de extensión .scm en la ventana de edición.

Save definitions, Save definitions as...: almacena las definiciones en un archivo.

- Edit: Opciones clásicas de portapapeles y de búsqueda y sustitución.
- View: Muestra u oculta ventanas del entorno o los números de línea del editor.
- Language: Nos permite fijar el nivel de lenguaje a utilizar entre varios posibles.
- Racket:

Run: Evalúa las definiciones.

Ask the program to quit: Interrumpe la evaluación actual.

Force the program to quit: Interrumpe inmediatamente la evaluación actual.

Reindent: sangra la selección actual.

Reindent all: sangra todo el texto.

Comment Out with Semicolons: comenta la selección actual (; comentario...)

Uncomment: elimina los comentarios de la selección actual.

- Scripts
- Tabs: Nos permite movernos entre las pestañas de edición abiertas.
- Help

Botonera

Por medio de estos botones accedemos a un conjunto de acciones sobre la ventana de edición, de izquierda a derecha tenemos:

- *Nombre de Archivo*: muestra el nombre del archivo actualmente en edición, al pulsarlo nos da la ruta completa y permite cargar archivos desde cualquiera de las carpetas de la ruta.
- (define ...): Nos permite localizar la definición de una función en el archivo en edición, y ordenarlas por posición en el archivo o alfabéticamente.
- *Salvar*: Nos permite guardar en disco los cambios realizados en el editor.
- *Check Syntax*: Analiza sintácticamente las definiciones introducidas. Nos indica los errores detectados, y si todo es correcto resalta palabras clave y funciones definidas. Al mover el ratón sobre nombres de funciones y argumentos relaciona por medio de flechas las definiciones y sus usos.
- *Debug*: depuración del código paso a paso.
- *Run*: Evalúa el programa del editor, haciendo disponibles sus definiciones en la consola.
- *Stop*: interrumpe la evaluación actual introducida en la consola, o emite un pitido sino está evaluando. Una sola pulsación solicita la cancelación la evaluación, mientras que una doble pulsación fuerza la cancelación inmediata.

Ventana de definiciones (editor)

Racket nos ofrece este editor donde introducir las definiciones de usuario. Su contenido puede ser tecleado directamente en la ventana o bien cargado desde disco. Este editor es un control estándar de edición de Windows, pero además incluye algunas características propias de *Scheme* que le convierten en un editor sensible al contexto: paréntesis eléctricos, auto-sangrado y resalte de bloques (secuencia de elementos entre paréntesis, corchetes o llaves).

Racket incorpora paréntesis eléctricos, que actúan cuando cerramos un bloque o una cadena entre comillas dobles, indicándonos cual es el carácter que abre dicho bloque moviendo el cursor a su posición y retornando de nuevo a la posición actual. Además, al cambiar de línea en el editor, automáticamente sangra la línea de acuerdo con la expresión en la que nos encontremos.

Ventana interactiva (consola del intérprete)

Esta ventana es la consola de comandos de *Scheme*, desde ella podemos llamar a cualquier función ya definida, cargar un archivo de definiciones (*load "<nombre_archivo>.scm"*) o incluso definir nuevas funciones por medio de (*define ...*).

Se pueden recuperar las entradas previas mediante `<Escape>-P` y también se puede ir a la siguiente entrada mediante `<Escape>-N`. También se puede marcar una entrada anterior y pulsando la tecla `<Return>` copiarla a la entrada actual.

Línea de estado

En ella se nos informa, de izquierda a derecha:

- Del nivel del lenguaje que tenemos seleccionado.
- La posición del cursor dentro del archivo que se esté editando, en formato `fila:columna`.
- El siguiente recuadro muestra el símbolo de reciclaje mientras Racket esté reciclando sus recursos (por ejemplo, al liberar la memoria).
- Por último, un icono de un personaje que está estático o corriendo según se esté editando o ejecutando un programa.

Edición, depuración y ejecución de programas

Una sesión de trabajo se basa en introducir las definiciones funcionales en el editor y probar su ejecución desde la consola, guardando en un archivo las definiciones para su posterior utilización.

Cargar y guardar archivos

Racket utiliza archivos con extensión `.rkt` (`.scm`, `.ss`, `.sch` o `.s` son otras extensiones válidas), para nuestro uso serán simples archivos de texto.

La carga de un programa en el editor se realiza por medio de la opción `File|Open`. Para guardar en disco las definiciones se puede pulsar directamente en el botón *Salvar* o bien utilizar las opciones `File|Save Definitions` o `File|Save Definitions As...`

También es posible cargar una serie de definiciones sin necesidad de pasar por el editor mediante la función (*load "archivo"*).

Compilación

Para hacer disponibles las definiciones que se cargan o editan se debe pulsar sobre el botón *Run*. A partir de ese momento se pueden utilizar las nuevas funciones de igual forma que las predefinidas. Racket nos avisa con un mensaje de advertencia si tratamos de evaluar expresiones desde la consola sin haber evaluado las últimas modificaciones del editor.

Depuración

Racket nos permite seguir la traza paso a paso de la evaluación de una expresión. Para ello debemos introducir al final del editor la llamada, o llamadas, a evaluar y a continuación pulsar en el botón *Debug*. Se abrirá una nueva barra de botones que permitir realizar la ejecución paso a paso y dos ventanas adicionales *Stack* y *Variables*.

Evaluación

La evaluación se realiza desde la consola del intérprete. Desde ésta se puede llamar tanto a funciones predefinidas como a funciones ya compiladas, además de poder realizar definiciones en la misma consola mediante la función (*define ...*).