

TRABAJO PRÁCTICO Nº 2

RUBY: CADENAS, CICLOS Y MÓDULOS

- 1.- Construir una función *“controlar_password?”* que pida por teclado una cadena y la compare con una cadena ya almacenada. Si coinciden, la función debe devolver TRUE; en otro caso, devolver FALSE. Utilizando el resultado de la función mostrar el mensaje “Acceso Permitido” o “Acceso denegado” según corresponda.
- 2.- Modifique el programa anterior para que sólo se permitan 3 intentos de clave no válida.
- 3.- Escriba una función *“inverso”* que invierta una cadena de caracteres previamente ingresada. (No se permite utilizar el método *reverse* de Ruby).
- 4.- Diseñe un programa que permita ingresar una cadena de no menos de 8 caracteres. Como salida transforme la cadena original agregando letras del alfabeto al azar entre las letras originales de la cadena. P/ej. REDOBLANTE => RAEBDCODBELFAGNHTIEJ
- 5.- Escriba una función llamada *“existe_caracter”* que realice la búsqueda de un carácter en una cadena previamente ingresada. La cadena ingresada deberá tener más de 6 caract. Deberá retornar su posición si existe (o cero si no existe).
- 6.- Ingresar dos cadenas de igual tamaño, mayores a 5 caract. Volver a solicitar su ingreso si las cadenas no tienen igual tamaño. Luego, si la primera cadena tiene más vocales que la segunda, unir ambas cadenas en una nueva cadena, caso contrario generar una nueva cadena intercalando cada letra de la primera cadena con cada letra de la segunda cadena y así sucesivamente.
- 7.- Escriba un programa que permita ingresar una contraseña que debe cumplir las siguientes pautas, si las pautas no se cumplen, se debe solicitar su ingreso nuevamente, informando la falla.
 - a) Debe tener más de seis caracteres y menos de veinte.
 - b) Debe comenzar con una letra (a..z)
 - c) No se permiten espacios en blanco
 - d) Debe poseer al menos un número (0..9)
- 8.- Ídem ejercicio anterior para:
 - a) Debe tener más de seis caracteres y menos de veinte.
 - b) Debe comenzar con una letra mayúscula (A..Z).
 - c) No se permiten espacios en blanco
 - d) Debe poseer tres números (0..9)
 - e) Debe poseer al menos cuatro letras que no deben repetirse.
- 9.- Cree un programa que permita ingresar una palabra de más de 4 letras sin espacios. Además debe contener una función llamada *“palíndromo?”*. Un palíndromo es una palabra que se lee igual tanto desde izq. a der. como de der. a izq. (también se las llaman *capicúa*, p.ej. *“neuquen”*). La función debe retornar true o false.