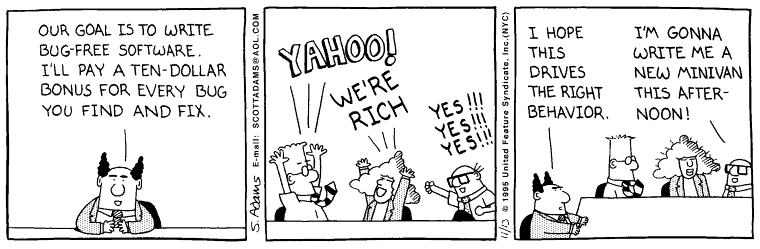
### Actividad de Reconocimiento - Las computadoras son como siervos fieles

#### Introducción

Pastrana Torres Victor Norberto

“*... Las computadoras son como siervos fieles, pues* ***ejecutan todo lo que les ordenamos****. Pero son todavía siervos sordos y mudos (y, a veces, tontos). Es difícil hacerles entender nuestras órdenes, hay que saber comunicarse con ellas en su idioma: algún lenguaje de programación…*” [1].



*Tira cómica* [*Dilbert por Scott Adams*](https://dilbert.com/)

De acuerdo a [2]:

“... los dispositivos de procesamiento de información deben de comprenderse en tres diferentes niveles antes de asumir que tales dispositivos se han realmente comprendido. En el extremo a más alto nivel, está la parte más abstracta que corresponde a la teoría computacional. En la parte central se encuentra la representación de entrada y salida y el algoritmo que se usará para transformar una en la otra. En el otro extremo se encuentran los detalles sobre cómo el algoritmo y la representación son físicamente desplegados - la arquitectura de cómputo detallada.

|  |  |  |
| --- | --- | --- |
| **Teoría computacional** | **Algoritmo** | **Implementación en Hardware** |
| ¿Cuál es la meta computacional, por qué es la apropiada y cuál es la lógica de la estrategia mediante la cual se puede llevar a cabo? | ¿Cómo puede esta teoría computacional puede ser implementada? En particular, ¿Cuál es la representación para la entrada y la salida y cuál es el algoritmo de transformación? | ¿Cómo se puede desplegar físicamente la representación y el algoritmo? |

#### Importancia de la Teoría Computacional

La percepción subyacente depende más de los problemas computacionales que deben resolverse que del hardware particular en el que se implementan sus soluciones. Visto también de otro modo, es probable que un algoritmo se entienda más fácilmente al comprender la naturaleza del problema que se está resolviendo, que al examinar el mecanismo (el hardware) en el que está incorporado. Por ejemplo, no se puede aprender cómo vuelan las aves si se estudian sólo las plumas de las aves, es necesario entender de aerodinámica, sólo entonces cobra sentido el tipo de la pluma y el tamaño de las aves.”

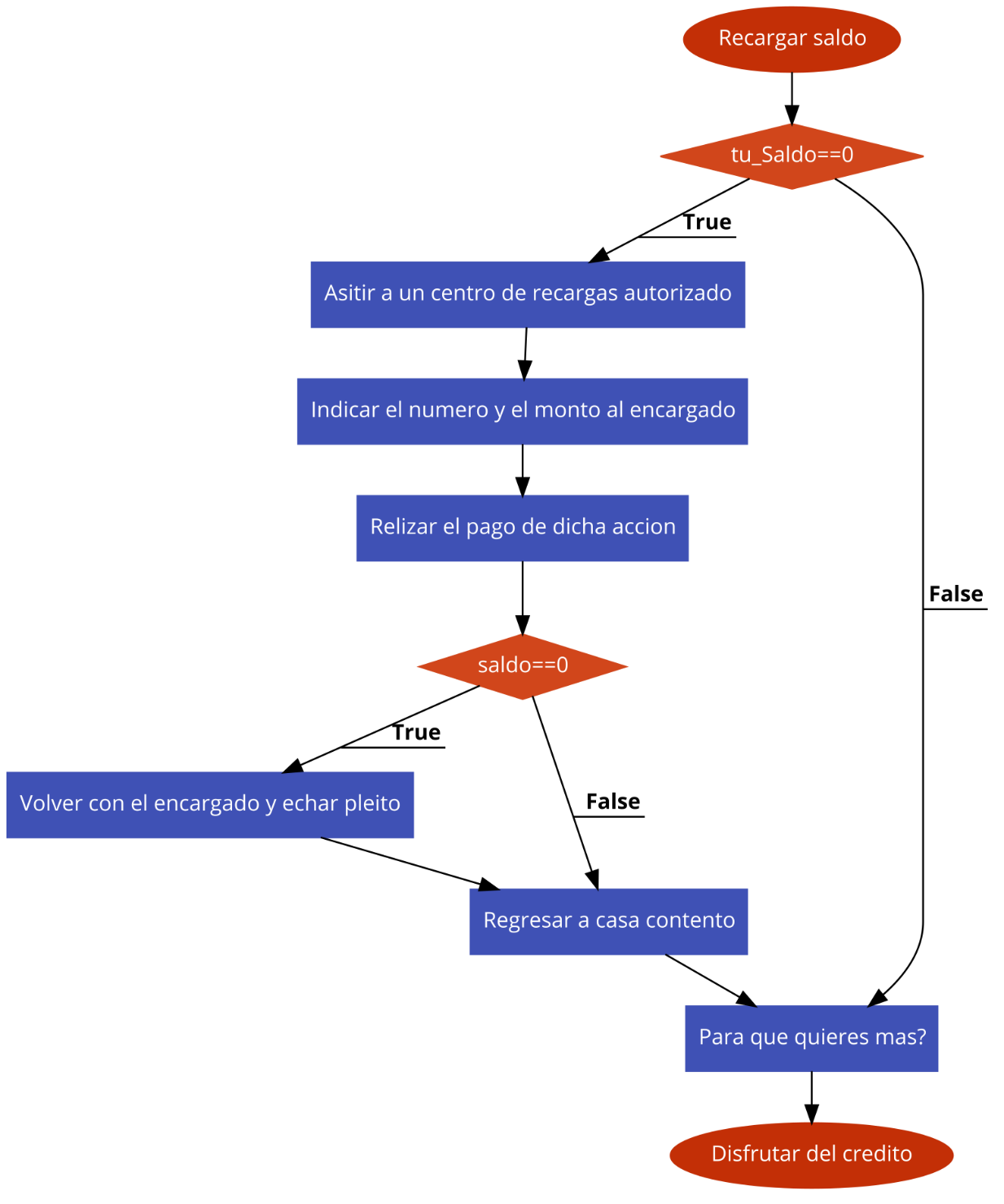
#### Instrucciones de la Actividad

1. Un lenguaje de programación es una notación para representar algoritmos y estructuras de datos. Escriba dos ventajas y dos desventajas de utilizar un lenguaje de programación como notación única durante todo el proceso de desarrollo.

|  |  |
| --- | --- |
| **Ventajas** | **Desventajas** |
| Compatibilidad entre versiones: al desarrollar un proyecto y defnir un lenguaje como predeterminado, no existiran problemas de conexion de modulos en un futuro  Implementación directa, al mostrar el desarrollo de un sistema directamente con el código, se tendra la posibilidad de realizar pruebas en tiempo real que si se utilizan únicamente modelos. | Dificultad de lectura: no todas la personas saben utilizar un lenguaje de programación  Conocimientos minimos, si se describe el desarrollo de un software por medio de un lenguaje de programacion se necesita que la persona que lo leerá tenga conocimentos previos del lenguaje. |

1. En el recuadro a continuación, escriba un *buen* algoritmo informático para explicar cómo recargar el crédito de su teléfono móvil. Asegúrese de utilizar un conjunto claro de instrucciones paso a paso.

|  |
| --- |
| Recargar saldo;  SI (tu\_Saldo==0){  Asitir a un centro de recargas autorizado;  Indicar el numero y el monto al encargado;  Relizar el pago de dicha accion;  Si(saldo==0){  Volver con el encargado y echar pleito;  }  Fin si  Regresar a casa contento;  }  Fin Si  Para que quires mas? 😐 ;  Disfrutar del credito; |

1. Vaya a la página [code2flow - interactive code to flowchart converter](https://code2flow.com/) elija probar la herramienta, después escriba el pseudocódigo del ejercicio 2. Obtendrá el [diagrama de flujo automáticamente](https://docs.google.com/document/d/1VGAVwLyUZcc6wpzzQBBvWIfR_8w2bRD0dzrq2PXBDXE/edit?usp=sharing). Incluya en el reporte las evidencias de este ejercicio.

* Haga un poco de investigación sobre herramientas como el **pseudocódigo** y los **diagramas de flujo**, en el recuadro emita su conclusión sobre por qué son útiles en el desarrollo de software.

|  |
| --- |
| El pseudocodigo es una forma de describir los pasos que seguirá un programa, tambien se dice que un algortimo se puede expresar mediante pseudocodigo.  El pseudocodigo o el diagra de flujo debe ser relizado de forma clara y no ambiagua de tal forma que cualquier persona tenga la posibilidad de leerlo y entenderlo.  Los diagramas de flujo muestran el compprtamineto que seguira un programa. |

1. Escriba en el siguiente recuadro que hace cada línea del script de la figura. Realice la prueba de escritorio, esto implica que ANTES de ejecutar el script se evalúe manualmente el resultado de las instrucciones y responda la siguiente pregunta:

* ¿Qué se dibuja con el código?

|  |
| --- |
| Aqui se inicia la animación  En esta sección se especifica la posición final La dirección en al que se realizará el movimiento  Bajará el lapíz  Las veces que se repetirtan las acciones siguientes   El objeto realizá un cambio de posición De nueva cuenta hay un giro  Este es el tiempo de espera para que se repita el bloque previo |

* Ahora, utilice Scratch [Scratch - Imagine, Program, Share](https://scratch.mit.edu/) para transcribir el script (encontrará las instrucciones básicas para acceder a Scratch [aquí](https://docs.google.com/document/d/1qP6FpN4DoG9FzPnO5F9pKB7BzWsBbHq23g8b_4hYS4c/edit?usp=sharing)). Copie su bloque de códigos en el siguiente recuadro, salve además en el sitio de Scratch y descargue. Responda también en el recuadro la siguientes pregunta:
  + ¿Coincide lo que ejecutó con lo que escribió en el resultado?

En gran parte sí, aunque en el bloque de lapiz no pude configurarlo por completo, pero descartando este detalle en gran parte se predijo lo que realizaria el bloque de codigo.

|  |
| --- |
|  |

1. **Code&Fix**, revise el siguiente algoritmo [6]:

|  |
| --- |
| CODE  FIX  REPEAT |

**Code**.- Programar intentando que el resultado proporcione los objetivos deseados.

**Fix**.- Corregir el código para que funcione (normalmente remedando y añadiendo más código).

Responda con Falso o Verdadero de acuerdo a las siguientes afirmaciones.

|  |  |
| --- | --- |
| 1. Un efecto negativo de Code&Fix es que favorece a que la calidad del producto sea baja. | V |
| 1. Los sistemas a menudo terminan siendo código desordenado y descoordinado que carece de adaptabilidad, capacidad de reutilización e interoperabilidad. | V |
| 1. Los sistemas desarrollados *ad hoc* son difíciles de ampliar y mantener. | V |
| 1. Un efecto negativo de Code&Fix es que favorecen a que los sistemas sean demasiado complicados y tienden a tener escasa escalabilidad. | F |
| 1. Code&Fix es un modelo poco útil, pero bastante común. | V |
| 1. El código resultante de Code&Fix podría funcionar, pero es altamente probable de que adolece de problemas de arquitectura, nula escalabilidad, capacidad de mantenimiento, por mencionar algunas. | V |
| 1. Code&Fix es la práctica detrás de eXtreme Programming. | V |
| 1. Code&Fix es una buena forma de trabajar en un concurso de programación en que se debe desarrollar una aplicación en tan sólo 6 horas. | V |

### Referencias

[1] Alexander Gelbukh y Grigori Sidorov, Procesamiento automático del español

con enfoque en recursos léxicos grandes, Segunda edición, IPN 2010.

[2] [David Marr](https://en.wikipedia.org/wiki/David_Marr_(neuroscientist)), Vision A Computational Investigation into the Human Representation and Processing of Visual Information, W.H. Freeman and Company, 1982.

[3] Crash course in programming, with Scratch, [christophe@pallier.org](mailto:christophe@pallier.org), <http://www.pallier.org/lectures/AIP2015/01_Starting-from-Scratch/Starting-from-Scratch-doc.html#starting-fromwith-scratch>

[4] Lero The Irish Software Research Centre http://scratch.ie/home

[5] Curso Desarrollo de habilidades metacognitivas mediante el uso del coding, Televisión Educativa, México, Junio 2020.

[6] Code-and-fix, Marzo 2012, <http://calidadysoftware.blogspot.com/2012/03/code-and-fix.html>

[7] Stewart Baird, Sams teach yourself eXtreme Programming in 24 hour, Sams Publishing, 2003.