



## Análisis y diseño de algoritmos avanzados TC2038

Dr. Víctor de la Cueva

[vcueva@tec.mx](mailto:vcueva@tec.mx)

1

## Clase

---

- Días: Lunes y Jueves
- Horario: 9 a 11 hrs
- Salón: 5402

2

## Datos

---

- Dr. Víctor de la Cueva
- Aulas 1, Nivel 5
- e-mail: [vcueva@tec.mx](mailto:vcueva@tec.mx)
- Asesoría: por cita
  - Me escriben dándome 3 opciones de horarios.
  - Les contesto en cuál puedo.

3

## Contenidos de aprendizaje

---

- |                                      |                   |
|--------------------------------------|-------------------|
| • Conceptuales                       | • Procedimentales |
| 1. Técnicas de diseño de algoritmos. | – Ver en Canvas   |
| 2. Manejo de Strings.                | • Actitudinales   |
| 3. Grafos.                           | – Ver en Canvas   |
| 4. Geometría computacional.          |                   |
| 5. Técnicas de búsqueda avanzada.    |                   |

4

# Competencias

---

- Competencias de área
  - SICT0100. Fundamentación de sistemas computacionales
    - SICT0101C. Explica el funcionamiento de sistemas computacionales.
    - SICT0400C. Compromiso con la sustentabilidad
- Competencias disciplinares
  - STC0100. Desarrollo de Algoritmos Computacionales
    - STC0101B. Implementa algoritmos
    - STC0102B. Optimiza algoritmos
- Competencias transversales
  - SEG0500. Razonamiento para la Complejidad
    - SEG0502B. Pensamiento científico

5

# Competencias del curso

---

- SICT0101.C
  - Explica el funcionamiento de sistemas computacionales por medio de argumentaciones sustentadas en las interacciones entre los componentes y su entorno creando modelos conceptuales donde se describan los componentes y la relación con su entorno.
- SICT0401.C
  - Aplica los estándares y normas propios de su profesión contrastándolos contra las restricciones de uso de acuerdo al proceso, producto o servicio donde se va a aplicar usando las normas y estándares más relevantes al dominio del problema que se va a resolver, distinguiendo claramente entre ambos.
- STC0101.B
  - Implementa algoritmos computacionales confiables y correctos que solucionan problemas
- STC0102.B
  - Optimiza algoritmos computacionales robustos y eficientes que se aplican en el desarrollo de soluciones.
- SEG0502.B
  - Pensamiento científico: Resuelve problemas e interrogantes de la realidad, a partir de metodologías válidas y confiables.

6

## Plan de Evaluación

---

- Actividades de aprendizaje
  - Programas **INDIVIDUALES** 40%
- 2 Exámenes (presenciales e **individuales**)
  - Intermedio 10%
  - (Ev) Final 20%
- Actividades CON evidencia de competencia
  - (Ev) Situación problema 1 **EQUIPO** 10%
  - (Ev) Situación problema 2 **EQUIPO** 10%
  - (Ev) Póster y reflexiones finales **EQUIPO** 10%
- TOTAL 100%

7

## Fechas

---

- Prácticas de verificación
  - Lunes 26 de septiembre (material hasta la semana anterior)
  - Lunes 27 de noviembre (todo el material)
- Asuetos:
  - Lunes 20 de noviembre
- Semanas Tec
  - 6, 11-15 septiembre
  - 12, 23-27 octubre
  - 18, 4-8 diciembre
- Último día de clases: viernes 1 de diciembre (más la semana 18)

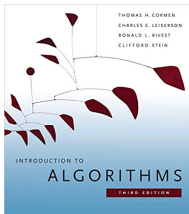
8

# Texto



- V.M. de la Cueva, L.H. González y P. Pérez.  
Algoritmos: Análisis, Diseño e Implementación.  
Editorial Digital Tecnológico de Monterrey (2021).

- Amazon: <https://www.amazon.com/-/es/Luis-Humberto-Gonz%C3%A1lez-Guerra-ebook/dp/B09SBV732P/>
- Apple Books: <https://books.apple.com/us/book/algoritmos-an%C3%A1lisis-dise%C3%B1o-e-implementaci%C3%B3n/id1609684135>
- Google Play: [https://play.google.com/store/books/details/Luis\\_Humberto\\_Gonz%C3%A1lez\\_Guerra\\_Algoritmos\\_an%C3%A1lisis-dise%C3%B1o-e-implementaci%C3%B3n?id=9dYFAA20BA&hl=es&gl=US](https://play.google.com/store/books/details/Luis_Humberto_Gonz%C3%A1lez_Guerra_Algoritmos_an%C3%A1lisis-dise%C3%B1o-e-implementaci%C3%B3n?id=9dYFAA20BA&hl=es&gl=US)



- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein.  
Introduction to Algorithms. 3rd ed, MIT Press (2009).

9

## Programas y proyectos

- Los programas se entregarán en forma individual y los proyectos en equipo, a menos que se indique otra cosa.
- Serán realizados en C++.
- Deberán contener lo que se indique en cada actividad.
- El programa fuente deberá estar comentado.
- Todo debe subirse en la misma liga de Canvas, con archivos por separado o en un ZIP, de acuerdo a lo que se solicite.
  - Un programa tendrá un archivo de tipo cpp o h, con el programa en C++.
  - Un proyecto tendrá 2 archivos, el programa en C++ (con extensión cpp o h) y uno de documentación (con extensión “doc” o “pdf”)

10

## Entrega de programas y proyectos

---

- Todos los programas y proyectos deben entregarse por medio de Canvas. Esta entrega se debe hacer a más tardar en la fecha indica por el profesor a las 11:59 de ese día.
  - Por ningún motivo se aceptarán tareas entregadas por otro medio
- Re-entregas de actividades:
  - Se puede recalificar la actividad si se entregó en tiempo.
  - Se darán 10 días naturales después de que se dan comentarios para que se haga una nueva entrega.
  - El ajuste de la calificación de la re-entrega será de a lo más 30 puntos adicionales a la calificación original asignada.
- Los alumnos tienen dos días más para entregar su tarea con una penalización del 30% el primer día y de 60% el segundo día:
  - Esta entrega se realizará por medio del email y la hora límite de recepción será a las 11:59 de ese día (La hora tomada será la que marque el email en el inbox del profesor).

11

## Sobre los códigos de las tareas

---

- Recuerde: Los ITC no copiamos nada, mucho menos código.
- Los códigos de las tareas deberán ser propios del alumno (generados completamente por él).
- Se pueden consultar fuentes en internet, las cuales se pondrán en las referencias, pero de ninguna forma se puede copiar el código.
- Tampoco se puede copiar el código de los compañeros, de alumno de otras clases, de trabajos de otros semestres o de herramientas de IA.
- En caso de que se detecte copia, se aplicará el reglamento del Instituto.

12

## Conocimientos previos

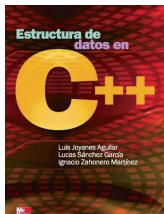
- Programación estructurada y POO
  - Estructuras de control: secuencia, decisión y repetición.
  - Modularidad: uso de funciones, parámetros por valor y variables.
  - Arreglos unidimensionales y bidimensionales.
  - Archivos de texto.
  - Recursividad.
  - Objetos: definición de clases, métodos, atributos, mensajes, etc.
  - String.
- Estructuras de datos :
  - Listas encadenadas
  - Pilas y filas.
  - Árboles binarios de búsqueda.
  - Hashing.
  - Grafos.
  - Algoritmos de ordenamiento.
- Matemáticas:
  - Límites
  - Series sencillas
  - Combinatoria básica
  - Álgebra matricial.

13

## Textos para conocimientos previos



- V.M. de la Cueva, L.H. González y E.G. Salinas. Estructuras de Datos y Algoritmos Fundamentales. Editorial Digital Tecnológico de Monterrey (2020).
  - Amazon: <https://www.amazon.com.mx/dp/B08FBJ9YFM>
  - Apple Books: <https://books.apple.com/us/book/id1526478309>
  - Google Play: <https://play.google.com/store/books/details?id=MXf1DwAAQBAJ>



- L. Joyanes, L. Sánchez e I. Zahonero. Estructura de datos en C++. McGraw-Hill (2007).

14

## Club de Programación

---

- Programación Competitiva (ICPC: <https://icpc.global/>)
- Reuniones los **miércoles**:
  - Horario por definirse
- Vamos a trabajar en forma de talleres
- Registro con Víctor de la Cueva ([vcueva@tec.mx](mailto:vcueva@tec.mx))
- Primera Reunión: **miércoles 23 de agosto**.
  - El horario y el salón se las mando por email a los registrados.

15

## Asociaciones

---



- Association for Computing Machinery (ACM), <http://www.acm.org/>.

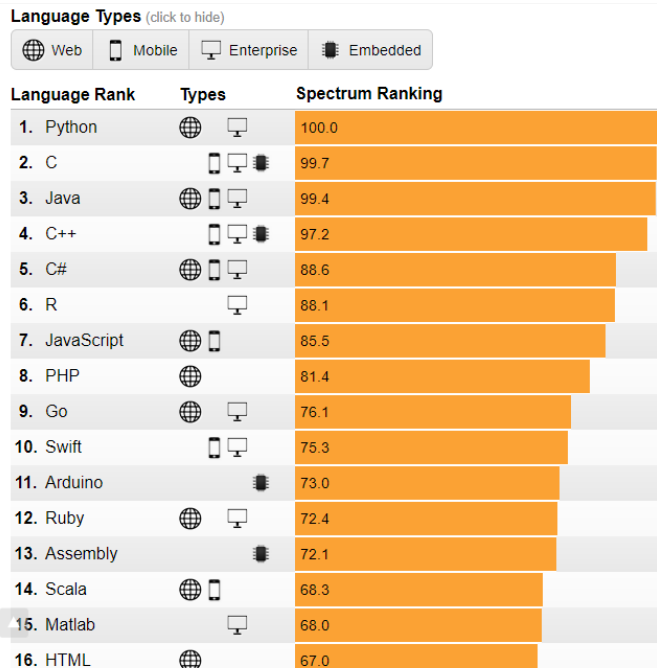


- Institute of Electrical and Electronics Engineers (IEEE), <http://www.ieee.org/>.

16

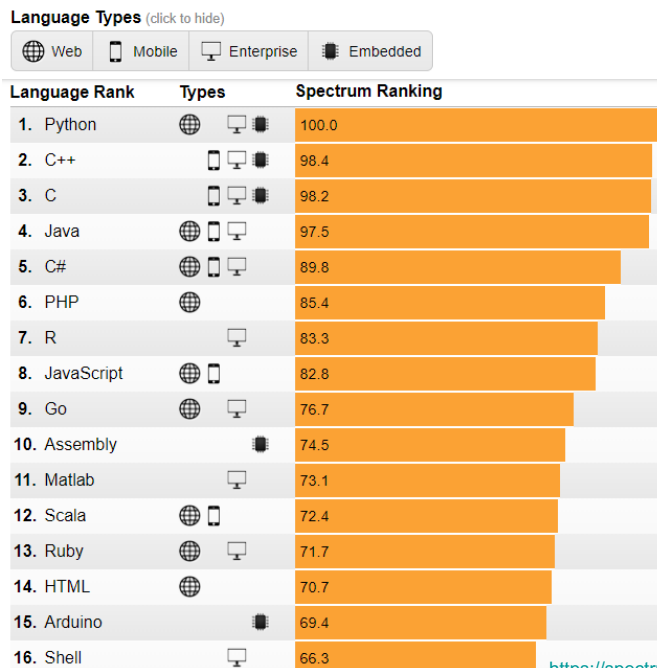


## Lenguajes del 2017 IEEE



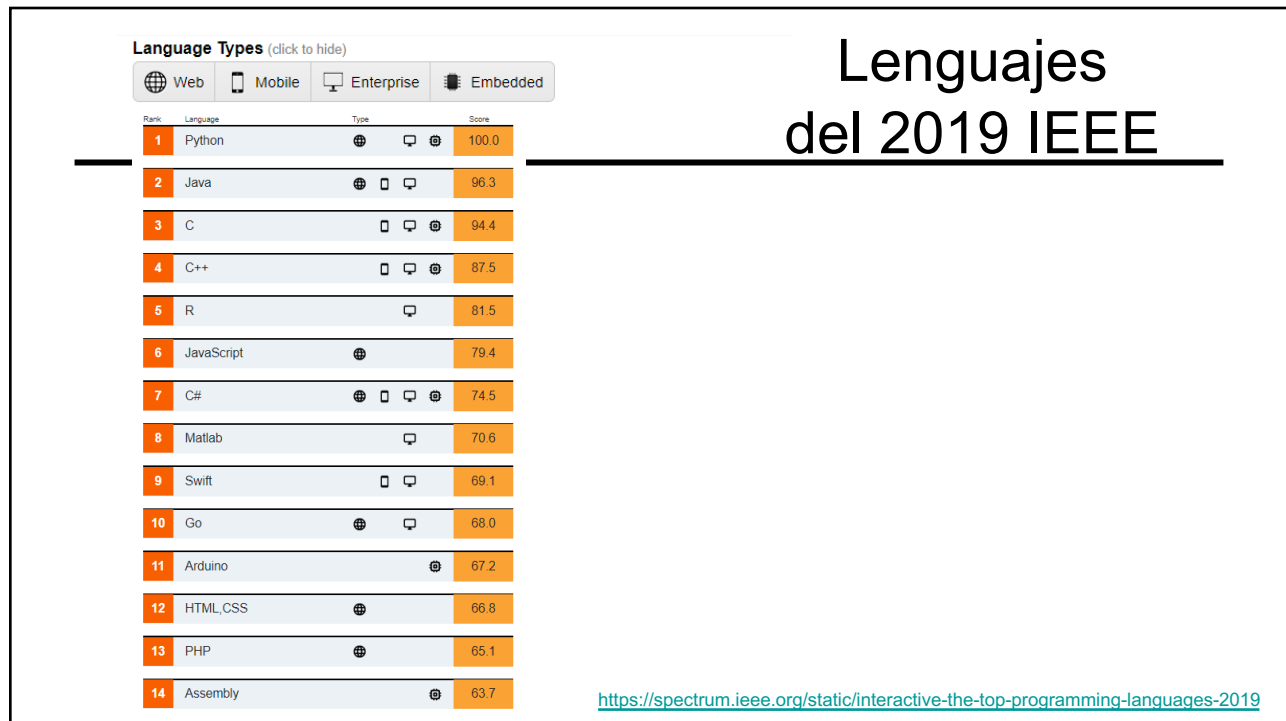
17

## Lenguajes del 2018 IEEE



<https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2018>

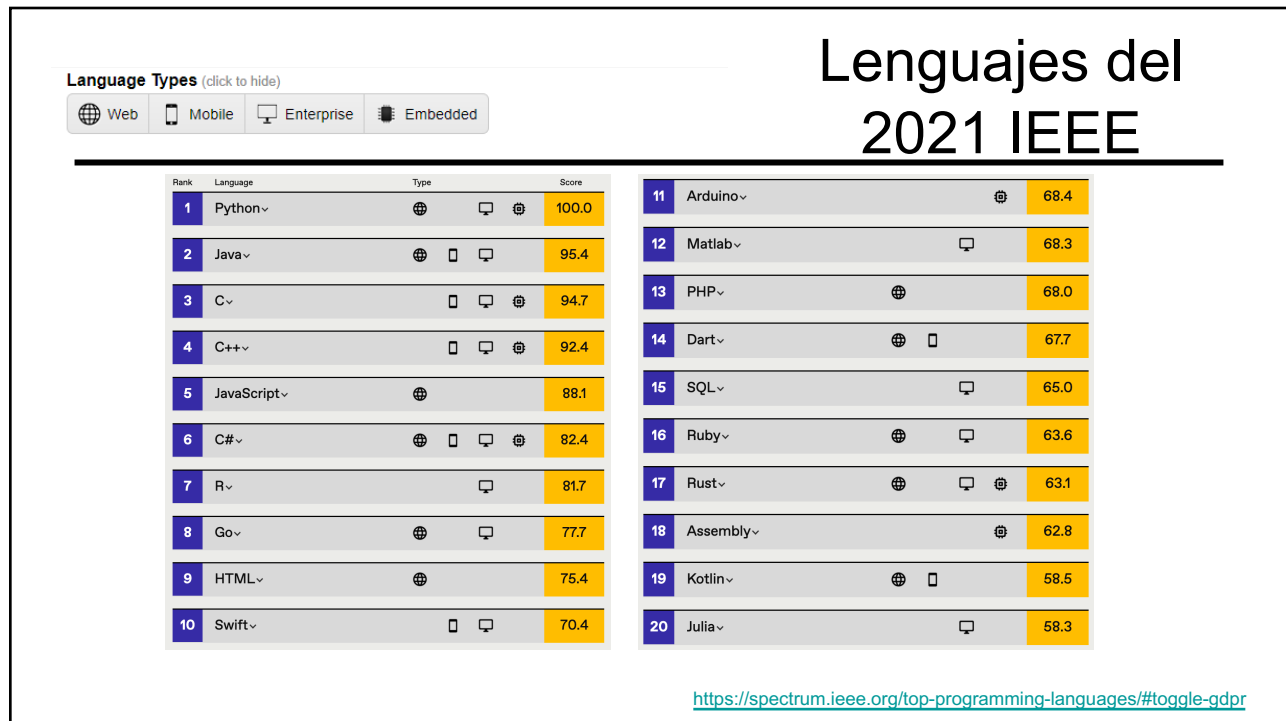
18



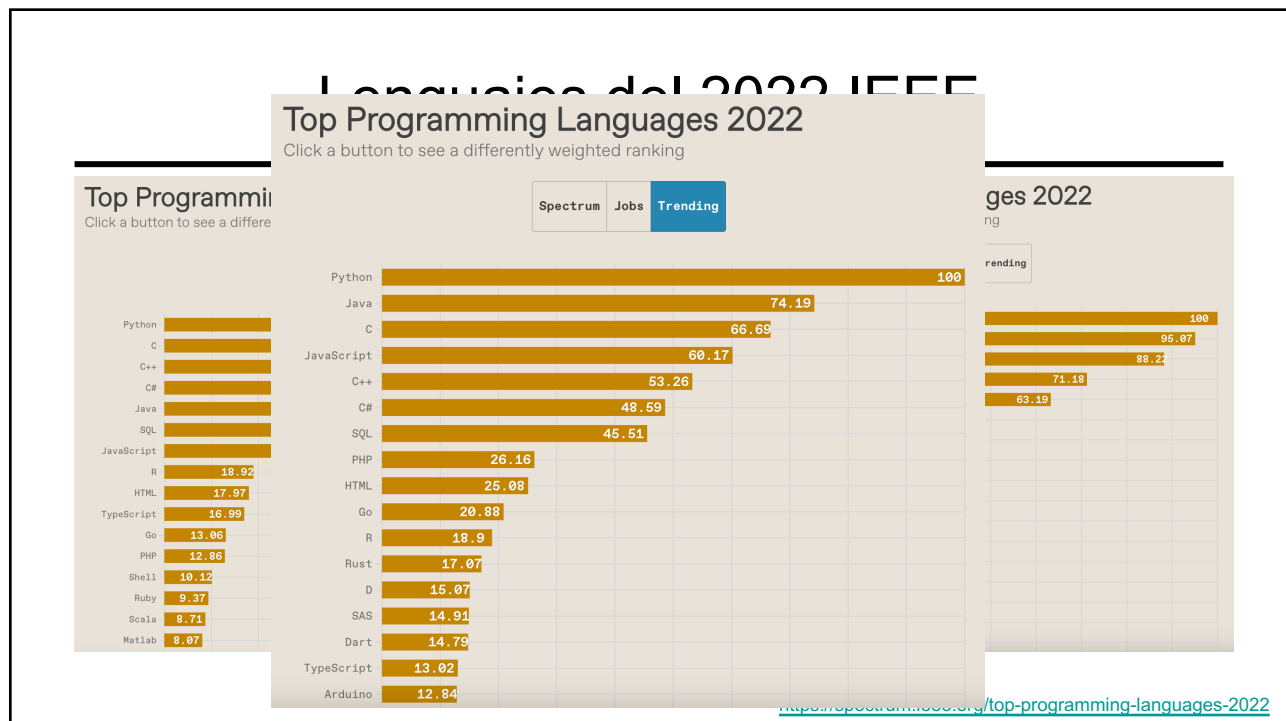
19



20



21



22

---

¿Dudas o  
comentarios?