

18/03/2019

TG1

**testing and developing**

Borja Ordóñez Bello, Javier Luque Moreno, Adrián López Godoy, Daniel Calin Stanus, Ignacio Burgos Lucha

**Scientific Cards**



Contenido

[1. Autores del trabajo, planificación y entrega 3](#_Toc3839640)

[1.1 Autores 3](#_Toc3839641)

[1.2 Planificación 3](#_Toc3839642)

[1.3 Entrega 3](#_Toc3839643)

[2. Descripción del tipo de tecnología 3](#_Toc3839644)

[3. Fuentes de información (documentos) 4](#_Toc3839645)

[3.1 Fuentes sobre el tipo de tecnología en general 4](#_Toc3839646)

[3.1.1 Fuente de información de CUDA 4](#_Toc3839647)

[3.1.2 Fuente de información de OpenCL 4](#_Toc3839648)

[3.1.3 Fuente de información de GPU 4](#_Toc3839649)

[3.2 Fuentes sobre CUDA 5](#_Toc3839650)

[3.2.1 Fuente de información 1 5](#_Toc3839651)

[3.2.2 Fuente de información 2 5](#_Toc3839652)

[3.2.3 Fuente de información 3 5](#_Toc3839653)

[3.3 Fuentes sobre tarjetas gráficas OpenCL 5](#_Toc3839654)

[3.3.1 Fuente de información 1 5](#_Toc3839655)

[3.3.2 Fuente de información 2 5](#_Toc3839656)

[3.3.3 Fuente de información 3 5](#_Toc3839657)

[4. Fuentes de información (cursos no gratuitos) 5](#_Toc3839658)

[4.1 Cursos no gratuitos sobre el tipo de tecnología en general 6](#_Toc3839659)

[4.1.1 Curso no gratuito 1 sobre el tipo de tecnología en general, *“Curso de Programación en GPU”*, hace una introducción general al curso de computación científica y cómo modelar usando CUDA. Es de modalidad presencial, en España, y tiene un coste de 4.100€. 6](#_Toc3839660)

[4.1.2 Curso no gratuito 2 sobre el tipo de tecnología en general, *“Numerical Computations with GPUs”*, curso en ebook con ISBN 978-3-319-06548-9, con coste de 99 USD: 6](#_Toc3839661)

[4.1.3 Curso no gratuito 3 sobre el tipo de tecnología en general, *“GPU Computing and Applications”,* curso en formato ebook con ISBN 978-981-287-134-3, con coste de 99 USD: 6](#_Toc3839662)

[4.2 Cursos no gratuitos sobre la tecnología específica A (CUDA) 6](#_Toc3839663)

[4.2.1 Curso no gratuito 1, impartido por la Universidad de Oxford. Para el público común, sin subsidios, tiene un costo de 2000 GBP. 6](#_Toc3839664)

[4.2.2 Curso no gratuito 2 sobre la tecnología específica A, localizado en Udemy, con nombre “CUDA programming Masterclass”, de modalidad online, impartido por K. Liyanage: 6](#_Toc3839665)

[4.2.3 Curso no gratuito 3 sobre la tecnología específica A, localizado en Udemy, con nombre *“Introduction to GPU Computing with CUDA”,* impartido por Orange Owl: 6](#_Toc3839666)

[4.3 Cursos no gratuitos sobre la tecnología específica B (OpenCL) 7](#_Toc3839667)

[4.3.1 Curso no gratuito 1 sobre la tecnología específica B, de modalidad presencial (USA), impartido por la empresa ArrayFire: 7](#_Toc3839668)

[4.3.2 Curso no gratuito 2 sobre la tecnología específica B *“Comprehensive OpenCL Programming for GPU and Multicore Architectures Info*”, impartido por Mindshare, con modalidad online: 7](#_Toc3839669)

[4.3.3 Curso no gratuito 3 sobre la tecnología específica B *“State-of-the-Art OpenGL with Qt”*, impartido por ICS services, en USA, con un costo de 2495 USD: 7](#_Toc3839670)

[5. Fuentes de información (cursos gratuitos) 7](#_Toc3839671)

[5.1 Cursos gratuitos sobre el tipo de tecnología en general 7](#_Toc3839672)

[5.1.1 Curso gratuito 1 sobre el tipo de tecnología en general, impartido en la Universidad de Oregon *(“CS 450 / 550 -- Introduction to Computer Graphics*”), que pone a la disposición de todo aquel que lo desee temario, recursos y ejercicios en el siguiente enlace: 7](#_Toc3839673)

[5.1.2 Curso gratuito 2 sobre el tipo de tecnología en general, *“Introduction to GPU Parallel Programming”,* de Donald Frederick, manual de 104 hojas, en formato pdf, disponible en la página web del Livermore High Performance Computing, un centro de cómputo público de Illinois: 7](#_Toc3839674)

[5.1.3 Curso gratuito 3 sobre el tipo de tecnología en general, *“GPU Computing: Quick Start”*, curso gratuito ofrecido por Wolfram Alpha: 7](#_Toc3839675)

[5.2 Cursos gratuitos sobre la tecnología específica A (CUDA) 8](#_Toc3839676)

[5.2.1 Curso gratuito 1 sobre la tecnología específica A *“Programming Massively Parallel Processors with CUDA”*, recursos online subidos a iTunes por la Universidad de Stanford: 8](#_Toc3839677)

[5.2.2 Curso gratuito 2 sobre la tecnología específica A, *“GPU Computing Using CUDA, Eclipse, and Java with JCuda”,* presente como ensayo de apoyo colgado en línea: 8](#_Toc3839678)

[5.2.3 Curso gratuito 3 sobre la tecnología especifica A, *“Udacity CS344: Intro to Parallel Programming”,* hospedado en Udacity y recomendado por Nvidia: 8](#_Toc3839679)

[5.3 Cursos gratuitos sobre la tecnología específica B (OpenCL) 8](#_Toc3839680)

[5.3.1 Curso gratuito 1 sobre la tecnología específica B, *“Hands On OpenCL”*, presente en Github, y creado por Simon Mcintosh-Smith, profesor de la Universidad de Bristol: 8](#_Toc3839681)

[5.3.2 Curso gratuito 2 sobre la tecnología específica B, *“Heterogeneous Parallel Programming”,* provisto por la Universidad de Illinois por medio de la plataforma 8](#_Toc3839682)

[5.3.3 Curso gratuito 3 sobre la tecnología especifica B*, “OpenCL”* de David W. Gohara, Ph.D. que imparte clase en el Center for Computacional Biology de la Washington University School of Medicine, impartido en forma de secuencia de vídeos en Youtube: 8](#_Toc3839683)

[ https://www.youtube.com/watch?v=QA483lIvL-4 8](#_Toc3839684)

[6. Ayudas económicas para estudiar las tecnologías 9](#_Toc3839685)

[7. Recursos para implementar las tecnologías 10](#_Toc3839686)

[7.1 Recursos para implementar la tecnología A (CUDA) 10](#_Toc3839687)

[7.1.1 En la página oficial de Nvidia, encontramos manuales para comenzar a aprender a programar en CUDA elaborados con la Universidad de Illinois, así como un ToolKit con todas las herramientas necesarias.Además, disponemos de herramientas como Jcuda, por si no queremos usar el IDE oficial de Nvidia: 10](#_Toc3839688)

[7.1.2 Recursos no gratuitos para implementar la tecnología A 10](#_Toc3839689)

[7.2 Recursos para implementar la tecnología B 10](#_Toc3839690)

[7.2.1 Encontramos un repositorio oficial de Github por parte de AMD con el SDK para trabajar con OpenGL, siendo un buen comienzo.También nos provee de una comunidad para resolver dudas, y un manual de programación inicial: 10](#_Toc3839691)

[7.2.2 Recursos no gratuitos para implementar la tecnología B 11](#_Toc3839692)

[8. Conclusiones 11](#_Toc3839693)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

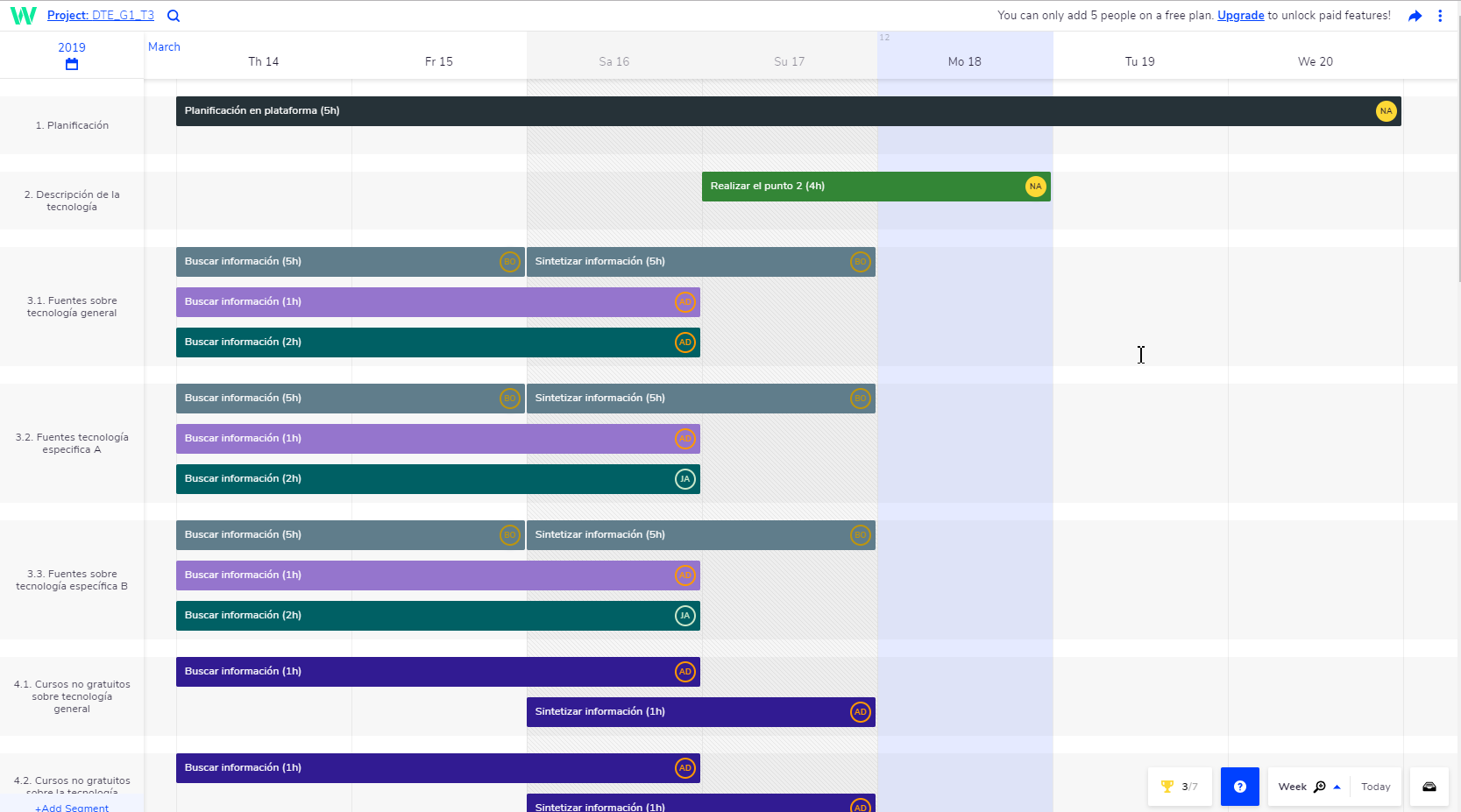
El presente documento ha sido elaborado por los siguientes estudiantes del Grado en Sistemas de la Información de la Universidad de Alcalá:

* Borja Ordóñez Bello
* Javier Luque Moreno
* Adrián López Godoy
* Daniel Stanus
* Ignacio Burgos Lucha

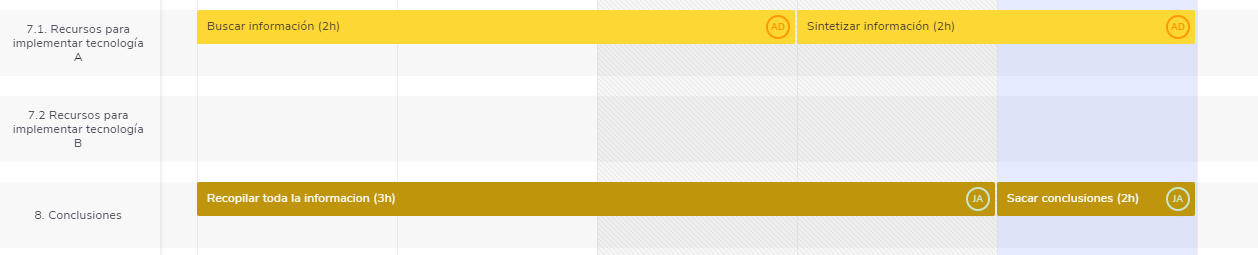
## 1.2 Planificación

Para organizar el trabajo hemos usado a la página de Teamweek, ya que es la que más claro y fácil nos parecía. Para consultar el diagrama la página genera un link: <https://app.teamweek.com/#pp/um7Hk_fVHIP9ezSn-96o3xp0WBkjr0_h>

Primero lo hemos organizado en los diferentes puntos y así todos participar en él.







Después nos hemos centrado en la planificación para las tareas de realización de documentos y gestión de reuniones.



## 1.3 Entrega

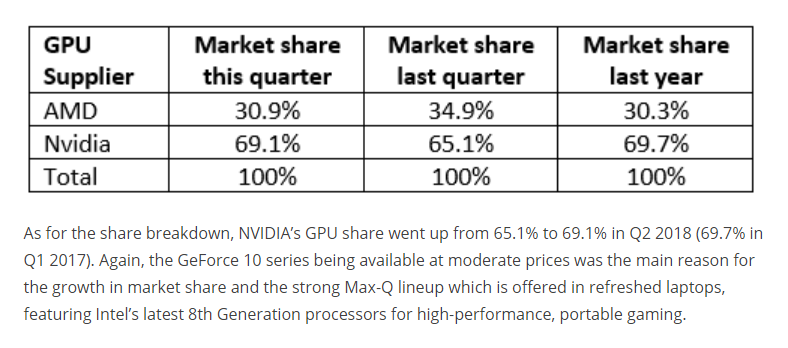
Nuestro enlace de GitHub para el trabajo grupal 1 (TG1) es el siguiente:

* <https://github.com/algoy/DTE/tree/master/TG1T3>

# 2. Descripción del tipo de tecnología

Nuestro proyecto versa sobre el uso de tarjetas gráficas en el mundo de la computación científica, y más concretamente, en aquellas tecnologías, incorporadas en las tarjetas o externas, que nos permiten aprovechar todo su poder de cómputo y eficiencia con fines científico-matemáticos, así como la investigación médica o diferentes modelados.

Siendo más concretos, investigaremos y expondremos resultados sobre las dos principales tecnologías que nos proporcionan los dos grandes fabricantes de tarjetas gráficas, que son Nvidia y AMD\*. Dichas tecnologías son Cuda, por para de Nvidia y OpenRadeonCompute por parte de AMD. Sin embargo, debido al estado de abandono de ORC, emplearemos la tecnología abierta OpenCL para testear la gráfica AMD.



**\*La fuente anterior muestra datos totales de porcentaje de mercado entre gráficas dedicadas (Nvidia y AMD). No se tiene en cuenta Intel, que es el mayor vendedor de gráficas, debido a sus soluciones integradas, mucho menos potentes.**

# 3. Fuentes de información (documentos)

## 3.1 Fuentes sobre gráficas en la computación

### 3.1.1 Fuente de información 1

Los cambios tecnológicos ocurridos en la computación grafica plantea muchos cambios. Por eso, este documento presenta una introducción básica y avanzada de la computación gráfica.

<http://sedici.unlp.edu.ar/handle/10915/24312>

### 3.1.2 Fuente de información 2

Documento que presenta una simulación gráfica de los efectos mecánicos de la computación de gráficos.

<https://www.researchgate.net/profile/Jan_Bender/publication/274940214_Position-Based_Simulation_Methods_in_Computer_Graphics/links/552cc4a40cf29b22c9c466df/Position-Based-Simulation-Methods-in-Computer-Graphics.pdf>

### 3.1.3 Fuente de información 3

Este trabajo presenta una guía para programar una tarjeta gráfica según los principios y reglas básicas de diseño.

<http://sedici.unlp.edu.ar/bitstream/handle/10915/41578/Documento_completo.pdf?sequence=1&isAllowed=y>

## 3.2 Fuentes sobre CUDA

### 3.2.1 Fuente de información 1

Introducción a la programación de GPU para tareas informáticas de propósito general. <https://computing.llnl.gov/tutorials/dataheroes/GPUParallelProgramming.pdf>

### 3.2.2 Fuente de información 2

Reducción de la complejidad de la programación de las GPU’s en base a CUDA-Lite.

<https://link.springer.com/chapter/10.1007/978-3-540-89740-8_1>

### 3.2.3 Fuente de información 3

El presente proyecto intenta comprobar la eficacia de las tecnologías GPGPU en su aplicación a algoritmos matriciales y vectoriales. Para ello se ha elegido el método Simplex Revisado como punto de partida, sobre el que se ha implementado una versión con ejecución pura en CPU junto con otra versión de ejecución mixta en CPU y GPU. De entre todas las tecnologías GPGPU existentes se ha elegido CUDA de la compañía NVIDIA.

<https://www.iit.comillas.edu/pfc/resumenes/4dee5b199a840.pdf>

## 3.3 Fuentes sobre tarjetas gráficas OpenCL

### 3.3.1 Fuente de información 1

Documento introductorio a OpenCL analizando el modelo de programación usando matrices.

<http://www-personal.umich.edu/~smeyer/cuda/Preso07-OpenCL.pdf>

### 3.3.2 Fuente de información 2

Relación de la CPU y GPU en cuanto a la programación mediante OpenCL.

<https://dl.acm.org/citation.cfm?id=2145863>

### 3.3.3 Fuente de información 3

Uso de OpenCL para gestionar la memoria compartida de procesos y recursos tales como la CPU y GPU.

<http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A628242&dswid=-8605>

# 4. Fuentes de información (cursos no gratuitos)

## 4.1 Cursos no gratuitos sobre el tipo de tecnología en general

### 4.1.1 Curso no gratuito 1 sobre el tipo de tecnología en general, *“Curso de Programación en GPU”*, hace una introducción general al curso de computación científica y cómo modelar usando CUDA. Es de modalidad presencial, en España, y tiene un coste de 4.100€.

* <https://www.nobleprog.es/cc/cudaprog>

### 4.1.2 Curso no gratuito 2 sobre el tipo de tecnología en general, *“Numerical Computations with GPUs”*, curso en ebook con ISBN 978-3-319-06548-9, con coste de 99 USD:

* <https://www.springer.com/la/book/9783319065472>

### 4.1.n Curso no gratuito n sobre el tipo de tecnología en general, *“GPU Computing and Applications”,* curso en formato ebook con ISBN 978-981-287-134-3, con coste de 99 USD:

* <https://www.springer.com/la/book/9789812871336>

## 4.2 Cursos no gratuitos sobre la tecnología específica A (CUDA)

### 4.2.1 Curso no gratuito 1, impartido por la Universidad de Oxford. Para el público común, sin subsidios, tiene un costo de 2000 GBP.

* <http://people.maths.ox.ac.uk/~gilesm/cuda/>

### 4.2.2 Curso no gratuito 2 sobre la tecnología específica A, localizado en Udemy, con nombre “CUDA programming Masterclass”, de modalidad online, impartido por K. Liyanage:

* <https://www.udemy.com/cuda-programming-masterclass/>

### 4.2.3 Curso no gratuito 3 sobre la tecnología específica A, localizado en Udemy, con nombre *“Introduction to GPU Computing with CUDA”,* impartido por Orange Owl:

* <https://www.udemy.com/introduction-to-gpu-computing-with-cuda/>

## 4.3 Cursos no gratuitos sobre la tecnología específica B (OpenCL)

### 4.3.1 Curso no gratuito 1 sobre la tecnología específica B, de modalidad presencial (USA), impartido por la empresa ArrayFire:

* <http://arrayfire.com/training/>

### 4.3.2 Curso no gratuito 2 sobre la tecnología específica B *“Comprehensive OpenCL Programming for GPU and Multicore Architectures Info*”, impartido por Mindshare, con modalidad online:

* <https://www.mindshare.com/Learn/OpenCL_Programming_for_GPU_and_Multicore_Architectures/Training>

### 4.3.3 Curso no gratuito 3 sobre la tecnología específica B *“State-of-the-Art OpenGL with Qt”*, impartido por ICS services, en USA, con un costo de 2495 USD:

* <https://www.ics.com/learning/training/state-art-opengl-qt>

# 5. Fuentes de información (cursos gratuitos)

## 5.1 Cursos gratuitos sobre el tipo de tecnología en general

### 5.1.1 Curso gratuito 1 sobre el tipo de tecnología en general, impartido en la Universidad de Oregon *(“CS 450 / 550 -- Introduction to Computer Graphics*”), que pone a la disposición de todo aquel que lo desee temario, recursos y ejercicios en el siguiente enlace:

* <http://web.engr.oregonstate.edu/~mjb/cs550/>

### 5.1.2 Curso gratuito 2 sobre el tipo de tecnología en general, *“Introduction to GPU Parallel Programming”,* de Donald Frederick, manual de 104 hojas, en formato pdf, disponible en la página web del Livermore High Performance Computing, un centro de cómputo público de Illinois:

* <https://computing.llnl.gov/tutorials/dataheroes/GPUParallelProgramming.pdf>

### 5.1.3 Curso gratuito 3 sobre el tipo de tecnología en general, *“GPU Computing: Quick Start”*, curso gratuito ofrecido por Wolfram Alpha:

* <https://www.wolfram.com/training/courses/hpc021.html>

## 5.2 Cursos gratuitos sobre la tecnología específica A (CUDA)

### 5.2.1 Curso gratuito 1 sobre la tecnología específica A *“Programming Massively Parallel Processors with CUDA”*, recursos online subidos a iTunes por la Universidad de Stanford:

* <https://itunes.apple.com/in/itunes-u/programming-massively-parallel/id384233322?mt=10>

### 5.2.2 Curso gratuito 2 sobre la tecnología específica A, *“GPU Computing Using CUDA, Eclipse, and Java with JCuda”,* presente como ensayo de apoyo colgado en línea:

* <https://www.codeproject.com/Articles/513265/GPU-Computing-Using-CUDA-Eclipse-and-Java-with-JCu>

### 5.2.3 Curso gratuito 3 sobre la tecnología especifica A, *“Udacity CS344: Intro to Parallel Programming”,* hospedado en Udacity y recomendado por Nvidia:

* <https://developer.nvidia.com/udacity-cs344-intro-parallel-programming>

## 5.3 Cursos gratuitos sobre la tecnología específica B (OpenCL)

### 5.3.1 Curso gratuito 1 sobre la tecnología específica B, *“Hands On OpenCL”*, presente en Github, y creado por Simon Mcintosh-Smith, profesor de la Universidad de Bristol:

* <https://handsonopencl.github.io/>

### 5.3.2 Curso gratuito 2 sobre la tecnología específica B, *“Heterogeneous Parallel Programming”,* provisto por la Universidad de Illinois por medio de la plataforma

* <https://www.class-central.com/course/coursera-heterogeneous-parallel-programming-427>

### 5.3.3 Curso gratuito 3 sobre la tecnología especifica B*, “OpenCL”* de David W. Gohara, Ph.D. que imparte clase en el Center for Computacional Biology de la Washington University School of Medicine, impartido en forma de secuencia de vídeos en Youtube:

### <https://www.youtube.com/watch?v=QA483lIvL-4>

# 6. Ayudas económicas para estudiar las tecnologías

Para obtener ayudas económicas a la hora de estudiar las tecnologías tendremos diferentes tipos distintos de ayudas económicas:

1. Becas
   * Como primera opción tendremos la beca general publica de estudios universitarios (becas MEC - Ministerio de Educación, Cultura y Deporte) El estado ofrece becas para estudios de Grado, estudios de Máster o estudios de Doctorado solo hay que cumplir con los requisitos que lo indican en el BOE o su página oficial.
   * Como segunda opción tenemos otro tipo de becas como ayudas por traslado si vives lejos de tu casa, el estado te podrá ofrecer y ayudar con una parte del dinero para que puedas financiarte el traslado, bien sea dándote una cantidad de dinero para el transporte o para la vivienda que tengas que pagar, ambas becas se podrán solicitar cada año.
   * Como tercera opción sería optar por becas del Instituto de la Juventud (INJUVE) que ofrecen distinto tipos de becas como: “Iniciación en la Empresa” y otras becas más.
   * Como cuarta opción podemos optar por ir al extranjero a estudiar ahí las tecnologías, tendríamos como opciones las becas erasmus o becas de la cámara de comercio, que son ayudas para estudiar en el extranjero.
2. Financiación

Otra de las formas de obtención de ayudas puede ser la presentación de un proyecto o idea basado en una tecnología a un concurso de startups para su financiación. A través de la creación de una aplicación basado en la tecnología CUDA o OpenCL y su presentación a uno de estos concursos, se puede obtener una financiación para seguir desarrollando e investigando dicha tecnología, ejemplo: www.ship2b.org.

También hay concursos que se realizan anualmente para la ayuda de startups como StartUp Programme organizado por PWC o BBVA Open Talent. En el ejemplo del primero, PWC pone a tu disposición varios consultores que te ayudan a diseñar un plan de negocio sólido al igual que varias sesiones de formación repartidas a lo largo de unos 6 meses. Al final de éstos, termina con la presentación del plan y un mínimo producto viable o demo del proyecto aplicación o negocio a concurso. Hay varios premios otorgados por diferentes organizaciones como la Comunidad de Madrid, Premio Enterprise Challenge, Premio PwC y otros de mínimo 1000€, con la oportunidad de continuar con la idea o el proyecto empezado.

# 7. Recursos para implementar las tecnologías

## 7.1 Recursos para implementar la tecnología A (CUDA)

### 7.1.1 En la página oficial de Nvidia, encontramos manuales para comenzar a aprender a programar en CUDA elaborados con la Universidad de Illinois, así como un ToolKit con todas las herramientas necesarias.Además, disponemos de herramientas como Jcuda, por si no queremos usar el IDE oficial de Nvidia:

* <https://developer.nvidia.com/educators/existing-courses>
* <https://developer.nvidia.com/cuda-downloads>
* <http://www.jcuda.org/>
* <https://developer.nvidia.com/nsight-eclipse-edition>
* <https://developer.nvidia.com/>

### 7.1.2 Recursos no gratuitos para implementar la tecnología A

Encontramos IDES de pago, que son los siguientes:

Microsoft Visual Studio 2017 Professional compatible con CUDA 9.0 y muy empleado en entornos laborales:

* <https://www.microsoft.com/es-es/p/visual-studio-professional-2017/dg7gmgf0dst5>

También será necesario contar con un equipo, portátil sobremesa o servidor, con tarjeta gráfica Nvidia que soporte la tecnología CUDA.

## 7.2 Recursos para implementar la tecnología B

### 7.2.1 Encontramos un repositorio oficial de Github por parte de AMD con el SDK para trabajar con OpenGL, siendo un buen comienzo.También nos provee de una comunidad para resolver dudas, y un manual de programación inicial:

* <https://github.com/GPUOpen-LibrariesAndSDKs/OCL-SDK/releases>
* <http://developer.amd.com/wordpress/media/2013/12/AMD_OpenCL_Programming_User_Guide2.pdf>
* <https://community.amd.com/community/devgurus/opencl>
* <https://developer.amd.com/tools-and-sdks/>

### 7.2.2 Recursos no gratuitos para implementar la tecnología B

Encontramos IDES de pago al igual que en el caso de CUDA, ya que las tecnologías en sí y sus librerías son de libre acceso.

Microsoft Visual Studio 2017 Professional es compatible con OpenCL y muy empleado en entornos laborales:

* <https://www.microsoft.com/es-es/p/visual-studio-professional-2017/dg7gmgf0dst5>
* <https://marketplace.visualstudio.com/items?itemName=galarius.vscode-opencl>

También será necesario contar con un equipo, portátil sobremesa o servidor, con tarjeta gráfica AMD/ATI que soporte la tecnología OpenCL.

# 8. Conclusiones

Las nuevas tecnologías GPGPU demuestra que las gráficas no sirven solo para jugar, sino que, gracias a su gran potencia de cómputo y eficiencia relativa, son una gran herramienta de cara a el futuro de la investigación científica. La existencia de alternativas libres a CUDA como OpenGL/OpenCL, nos permite desarrollar en esta tecnología sin límites de cara al futuro.

Una muestra de ello es que AMD ha abandonado su tecnología,AMD Stream, a favor de OpenCL.