

Sesión práctica de computación con métodos ab-initio en Ciencia de Materiales

**Instituto de Ciencia de Materiales de la
Universidad de Valencia**

Curso de verano 2024

Alejandro Molina Sánchez
(alejandro.molina@uv.es)

Alberto García Cristóbal
(alberto.garcia@uv.es)

Introducción

El software ab-initio o de primeros principios permite resolver la ecuación de Schrödinger independiente del tiempo para materiales periódicos o moleculares. La principal característica y ventaja de los códigos ab-initio es la ausencia de parámetros arbitrarios para obtener el resultado. Introduciendo la estructura del material (especies atómicas y sus posiciones) tendremos acceso a muchas de sus propiedades físicas, químicas, ópticas, etc.

En esta sesión práctica de demostración, aprenderemos cómo utilizar el código *Quantum Espresso* en dos ejemplos de materiales 2-dimensionales, tales como el grafeno y el nitruro de boro hexagonal.

Tarea 0 – Visualización de la celda unidad de grafeno y nitruro de boro hexagonal

Entramos en la *Virtual Box* (configurado previamente como se explica en el documento adjunto a este) y desde ahí entramos al directorio con los archivos para realizar las simulaciones, escribiendo en la terminal:

```
cd CursolCMUV2024
```

El comando *cd* nos permite entrar al directorio que escojamos. Si ahora escribimos el comando:

```
ls
```

podemos examinar los ficheros existentes en dicho directorio. Ahora, entramos al directorio del grafeno o del nitruro de boro y procedemos a visualizar la estructura atómica. Para ello utilizaremos el programa *Xcrysden*, ejecutando el comando:

```
xcrysden --pwi graphene_scf.in | xcrysden --pwi bn_scf.in
```

Una vez comprobado que la celda unidad es la correcta para describir el material en cuestión, podemos visualizar y modificar el fichero con un editor de texto como “vi” o gedit, ejecutando el comando:

```
vi graphene_scf.in | vi bn_scf.in
```

Tarea 1 – Cálculo autoconsistente (Self-consistent field calculation)

En las diapositivas podemos ver un ejemplo de “input file” de *Quantum Espresso*. Para realizar la simulación hay que pasarle dicho “input file” al ejecutable, tal que (de aquí en adelante se pondrán los comandos de ejemplo para el BN, siendo equivalente para el grafeno con los “input files” correspondientes):

```
pw.x < bn.scf | tee log_scf
```

La función “tee” imprime la salida del programa en pantalla y en el fichero `log_scf`. Una vez realizado el cálculo, podemos extraer información del fichero de salida (“output file”) investigando el fichero de salida `log_scf` ejecutando el comando:

```
vi log_scf
```

Por ejemplo, la información básica de la estructura es (para el BN, al principio del archivo):

```
bravais-lattice index      =          4
lattice parameter (alat)   =      4.7000  a.u.
unit-cell volume           =      229.5660 (a.u.)^3
number of atoms/cell       =          2
number of atomic types     =          2
number of electrons         =       8.00
number of Kohn-Sham states =          4
kinetic-energy cutoff       =      40.0000  Ry
charge density cutoff      =      160.0000  Ry
```

La energía del material (al final del fichero)

```
total energy              =      -26.70619398  Ry
```

Para realizar este ejemplo, hemos escogido un parámetro de red de 4.7 a.u., el cual suponemos cercano al real que desconocemos. Lo mismo podemos intentar con el grafeno.

Tarea 2 – Optimización de la estructura cristalina

La mayoría de los códigos ofrecen la posibilidad de encontrar la estructura con el mínimo de energía. Es lo que se conoce como el proceso de

optimización de la celda unidad. Para ello, hemos preparado los ficheros “bn_relax.in” y “graphene_relax.in”. Las opciones que hemos añadido son:

```
&control  
    calculation = 'vc-relax'  
  
&ions  
    ion_dynamics = 'bfgs'  
/&end  
&cell  
    cell_dofree = '2Dxy'  
    cell_dynamics = 'bfgs'  
/&end
```

Imponiendo el tipo de cálculo “*vc-relax*” seleccionamos dos opciones. Por un lado, minimizar la energía respecto a las posiciones de los átomos dentro de la celda unidad, controlado por la variable “*ion_dynamics*”. Por otro lado, minimizamos la energía variando también la celda unidad, mediante la variable “*cell_dynamics*”. Como estamos trabajando con un material 2D, hemos seleccionado la opción “*cell_dofree = 2Dxy*”, la cual solo optimiza los parámetros de red en el plano, dejando invariable la distancia entre capas.

Podemos proceder tanto con el BN como con el grafeno. Hemos fijado para este ejemplo “*Ecutwfc = 40 Ry*” y un mallado de puntos de la zona de Brillouin de $K = [3\ 3\ 1]$. Además, introducimos un valor de 5.3 Bohr para el parámetro de red, con el fin de observar la variación del mismo durante el cálculo. Para realizar la simulación, escribimos:

```
pw.x < bn_relax.in | tee log_relax
```

y encontraremos para ambos casos los siguientes resultados

BN = 4.672647157 Bohr

Grafeno = 4.632656887 Bohr

Como podemos apreciar, los parámetros de red de ambos materiales son muy parecidos. Sin embargo, como veremos más adelante sus estructuras de bandas son radicalmente diferentes.

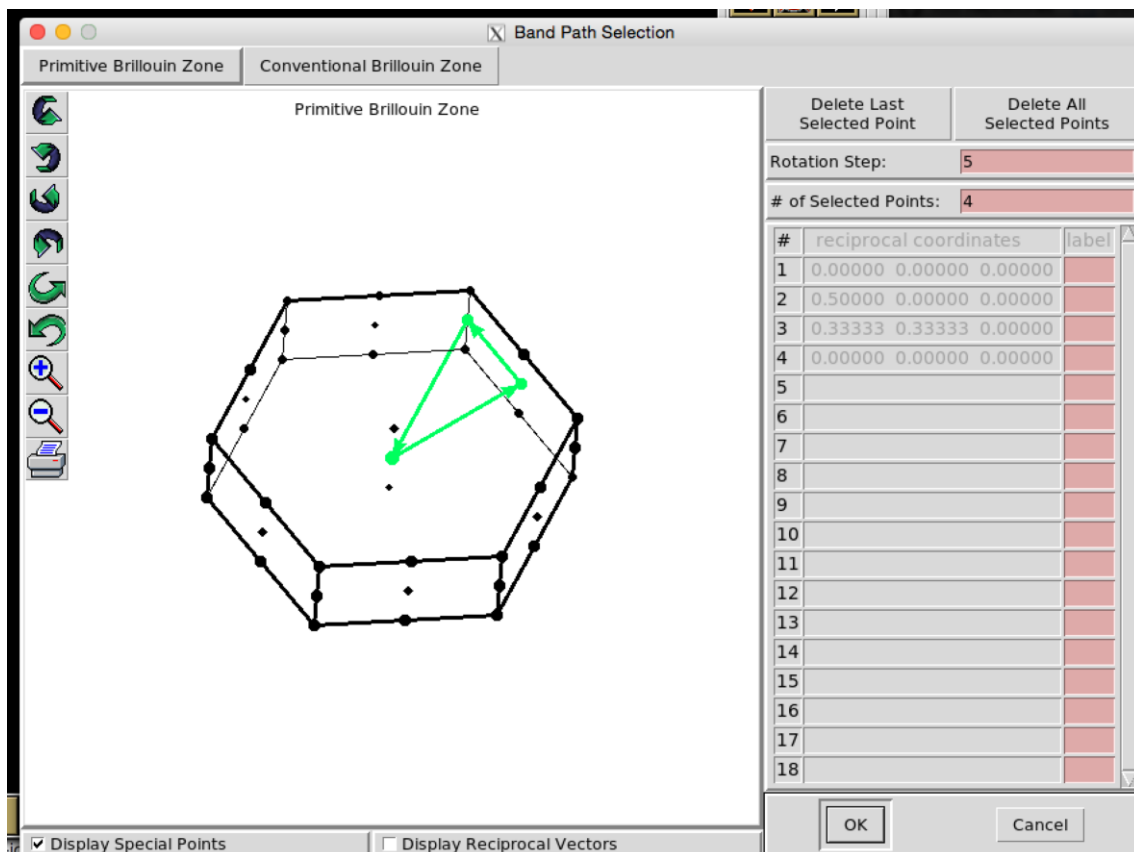
Otra tarea necesaria a la hora de optimizar el parámetro de red (o a la hora de calcular cualquier magnitud de interés) es evaluar su cambio cuando vamos aumentando el valor de los parámetros que describen el sistema. Si ahora incrementamos *Ecutwfc* en pasos de *10 Ry* o el mallado *k* a las dimensiones $[6\ 6\ 1]$ o $[12\ 12\ 1]$ veremos que el parámetro de red también cambia. Esto se llama convergencia y hay que realizarla siempre si queremos obtener resultados realistas o al menos consistentes.

Tarea 3 – Cálculo y representación de la estructura de bandas

Una vez realizado el cálculo del parámetro de red, podemos calcular la estructura de bandas. Esta tarea requiere de varios pasos:

- (a)– Calculamos la densidad electrónica seleccionando un cálculo “scf”, utilizando los parámetros optimizados. Para ello, debemos sustituir en el archivo de la *Tarea 1* el parámetro de red y las posiciones atómicas con las optimizadas. En el ejemplo que mostramos a continuación, utilizamos los parámetros optimizados obtenidos con un $E_{cutwfc} = 40 \text{ Ry}$ y $K = [6 \ 6 \ 1]$.
- (b)– Calculamos las energías y las funciones de onda a lo largo de un camino en la zona de Brillouin. Esto lo podemos encontrar en los ficheros “bn_bands.in” y “graphene_bands.in”. Hemos seleccionado el camino Gamma-M-K-Gamma a lo largo de la zona de Brillouin hexagonal mostrada en la figura. También hemos añadido más bandas con la variable “nbnd=10”. Para ello ejecutamos:

`pw.x < bn_bands.in | tee log_bands`



(c) – Por último, debemos realizar un post-procesado de los datos para que estos se puedan representar de forma sencilla por el usuario. Esto lo podemos realizar mediante los ficheros “*bn.bands_pp.in*” y “*graphene_bands_pp.in*”, los cuales resultarán en varios “*output files*”, entre ellos uno que termina en *.gnu*. En dicho archivo están guardadas las energías de cada banda así como las distancias que necesitamos para obtener la estructura de bandas. Para ello ejecutamos:

```
bands.x < bn_bands_pp.in | tee log_bands_pp
```

Una vez realizado estos tres pasos, estamos listos para representar el resultado. Podemos utilizar *gnuplot*, uno de los programas de representación que permite obtener una representación rápida y sencilla de los datos. Otra opción plausible es utilizar *Python*.

Para echar un vistazo rápido a la estructura de bandas, en primer lugar, utilizamos *gnuplot*, al cual accederemos escribiendo el comando:

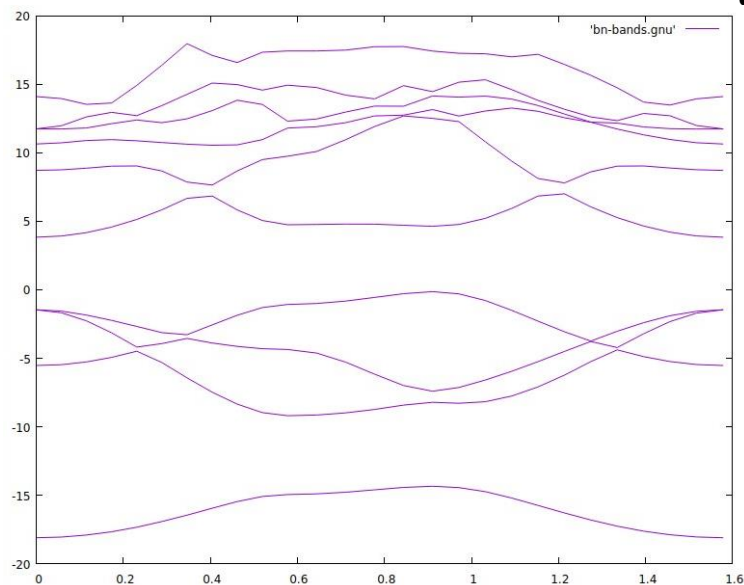
gnuplot

en la terminal. Una vez dentro, escribiremos:

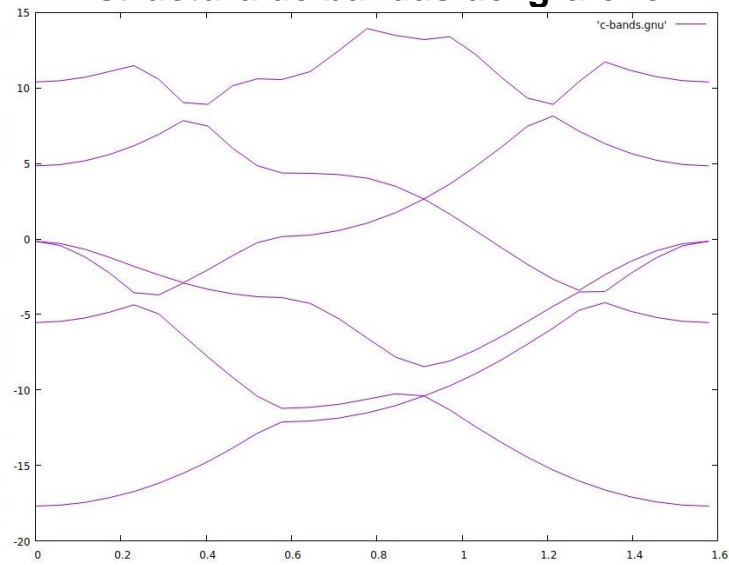
```
plot 'bn-bands.gnu' w l (de gnuplot se sale escribiendo q en la terminal)
```

y obtendremos un resultado como el siguiente:

Estructura de bandas del nitruro de boro hexagonal



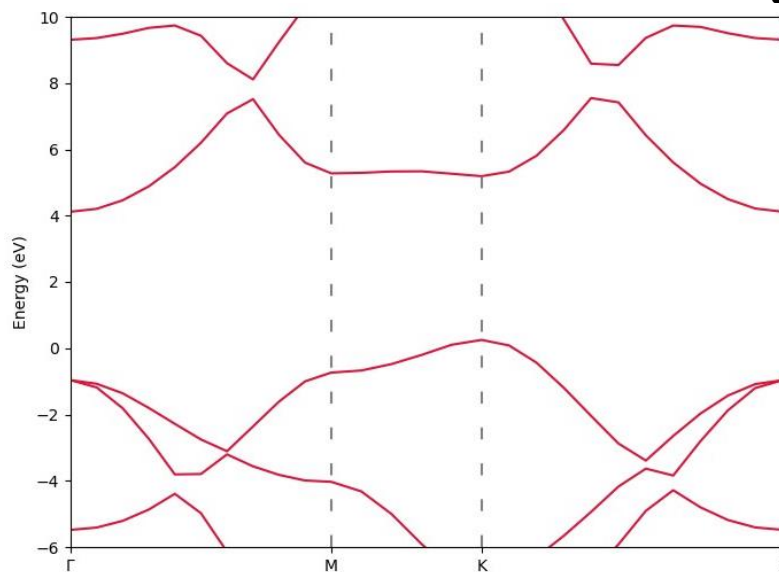
Estructura de bandas del grafeno



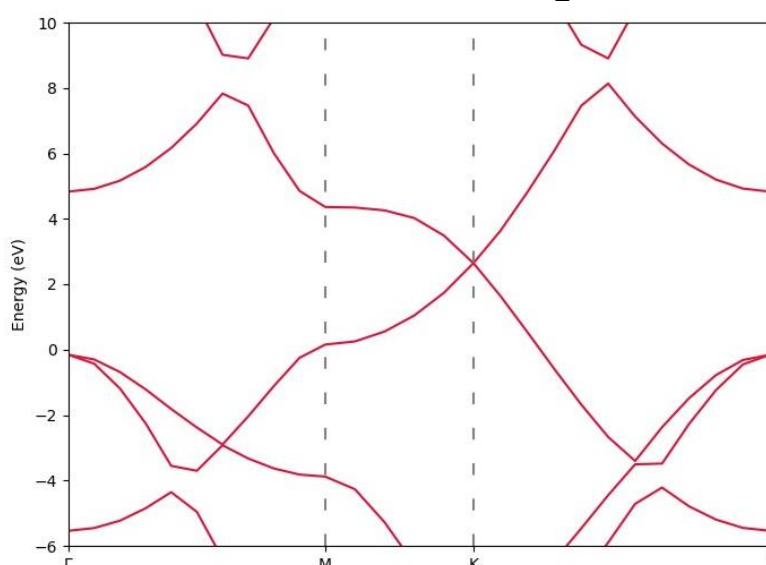
También podemos representar los datos utilizando *Python* mediante el script previamente preparado. Dicho script se ha preparado para que muestre los resultados en un formato más aseado, similar a como puede aparecer una estructura de bandas en un artículo científico. Para ello, escribimos el comando:

`python plot_bands.py`

Estructura de bandas del nitruro de boro hexagonal



Estructura de bandas del grafeno



Tarea 4 – Cálculo y representación de las funciones de onda

Por último, vamos a calcular y representar las funciones de onda en el espacio real para el grafeno y BN. Mediante este cálculo podemos entender dos puntos importantes. Por un lado, la forma de la función de onda, y por otro, dónde se encuentran dichas funciones de onda. Esto último nos permite entender que si, por ejemplo, la función de onda de la banda de valencia en el punto Gamma está sobre el boro, quiere decir que la banda en esa región de la zona de Brillouin está principalmente constituida por los electrones del boro. Esto se puede extender a toda la estructura de bandas, lo que nos permite obtener un entendimiento mayor sobre las propiedades electrónicas del material.

En este cálculo se realiza un post-procesado de la densidad electrónica calculada en el “scf”. Para esta parte utilizaremos los ficheros “pp-wf-vb.in” y “pp-wf-cb.in” tanto en grafeno como en BN, para calcular la función de onda en la banda de valencia y conducción, respectivamente. Primero volveremos a calcular el “scf” que ya hemos hecho anteriormente, añadiendo la línea “nbnd = 10”. Después, tomando el BN como ejemplo, simplemente debemos ejecutar el comando:

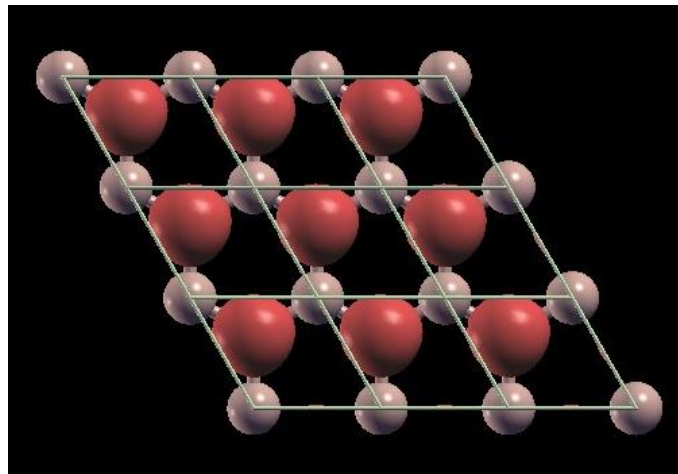
```
pp.x < pp-wf-vb.in | tee log_pp_vb
```

Esto producirá un archivo “wf-vb.xsf” el cual ha sido propiamente escrito para que pueda ser visualizado en *Xcrysden*. Para ello, escribimos en la terminal:

`xcrysden --xsf wf-vb.xsf`

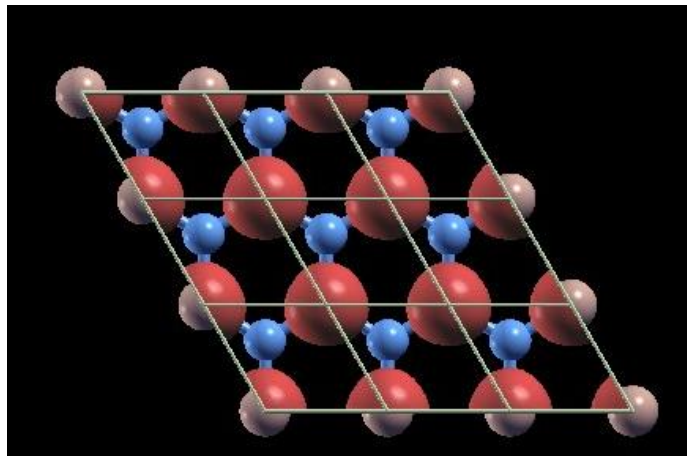
Esto mostrará la celda unidad como la hemos visto previamente. Para obtener una correcta visualización, primero accedemos a la pestaña “*Modify*” en la barra horizontal de arriba y pinchamos en “*Number of Units Drawn*”. Acto seguido, ponemos que la repetición de la dirección X y la dirección Y sean 3, posteriormente pinchamos “*Update*”. Finalmente, accedemos a la pestaña “*Tools*” y “*Data Grid*”. Esto identifica que hay una isosuperficie para representar. Pinchamos en “*OK*” y escribimos en la pestaña “*Isovalue*” 0.01, pinchamos en el cuadrado “*Render +/- isovalue*” y en el círculo “*to whole structure*”. Finalmente, pinchamos a “*submit*” y veremos la función de onda del BN en la banda de valencia en el punto de alta simetría “K”, ya que observando la estructura de bandas vemos que es donde se encuentra la transición directa de menor energía.

Función de onda de la banda de valencia del BN en K

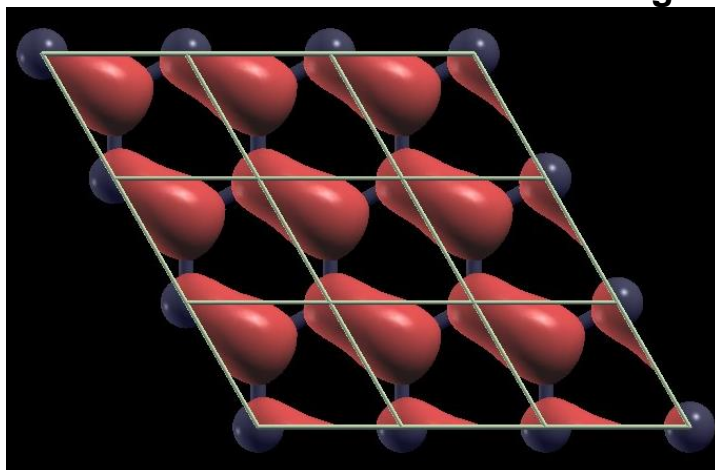


Realizando lo mismo para la banda de conducción del BN así como lo mismo para el grafeno, tenemos:

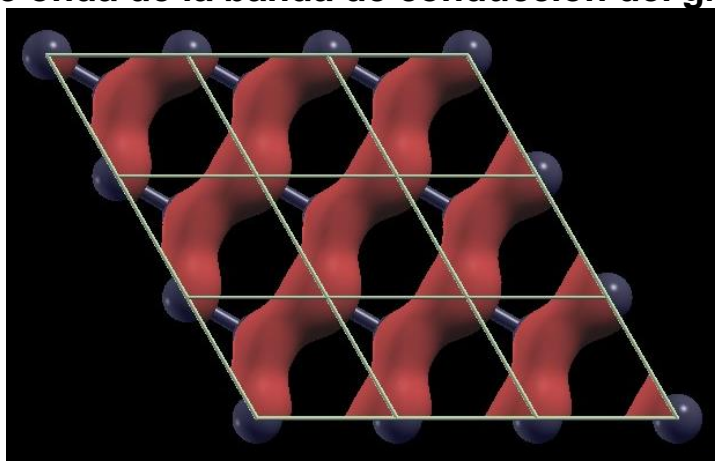
Función de onda de la banda de conducción del BN en K



Función de onda de la banda de valencia del grafeno en K



Función de onda de la banda de conducción del grafeno en K



Límite de los cálculos ab-initio

Podría parecer que con *Quantum Espresso* puede hacerse un cálculo de cualquier sistema. Aunque en teoría es así, a poco que empecemos a aumentar los parámetros (buscando convergencia o precisión) veremos que el tiempo de cálculo aumenta considerablemente. En resumen, el tiempo de cálculo depende del tamaño de la base de ondas planas y puntos k , del número de átomos y del tipo de funcional. Si aumentamos *Ecutwfc* junto con el mallado de puntos k veremos que pronto el ordenador tarda demasiado o incluso detiene el cálculo por falta de memoria.

Para poder realizar simulaciones de sistemas grandes y en un tiempo asumible, tenemos a nuestra disposición los superordenadores (ver referencias). La Universidad de Valencia dispone del suyo propio, *Tirant*.

Referencias

La documentación de *Quantum Espresso* así como muchos otros tutoriales se encuentra [aquí](#).

Otros programas de primeros principios son:

[ABINIT](#)

[VASP](#)

[FHI-AIMS](#)

Para interesados en códigos post-DFT (concretamente en teoría de perturbaciones de muchos cuerpos):

[YAMBO](#)

Información sobre supercomputadores:

[Tirant](#)

[BSC](#)

[Ministerio de Ciencia](#)