Laboratorio de física de partículas



Objetivos

Simular la interacción básica de partículas con ciertos medios gracias al *software* Geant4 (<https://geant4.web.cern.ch/>), desarrollado por el CERN (<https://home.cern/>).

En este laboratorio vamos a manejar una serie de contenedores Docker (<https://www.docker.com/>) para *orquestar* una serie de servicios sencillos *virtualizados*. En lugar de instalar Docker en nuestra computadora (que podríamos hacerlo perfectamente), vamos usar el servicio gratuito Play with Docker (<https://labs.play-with-docker.com/>).

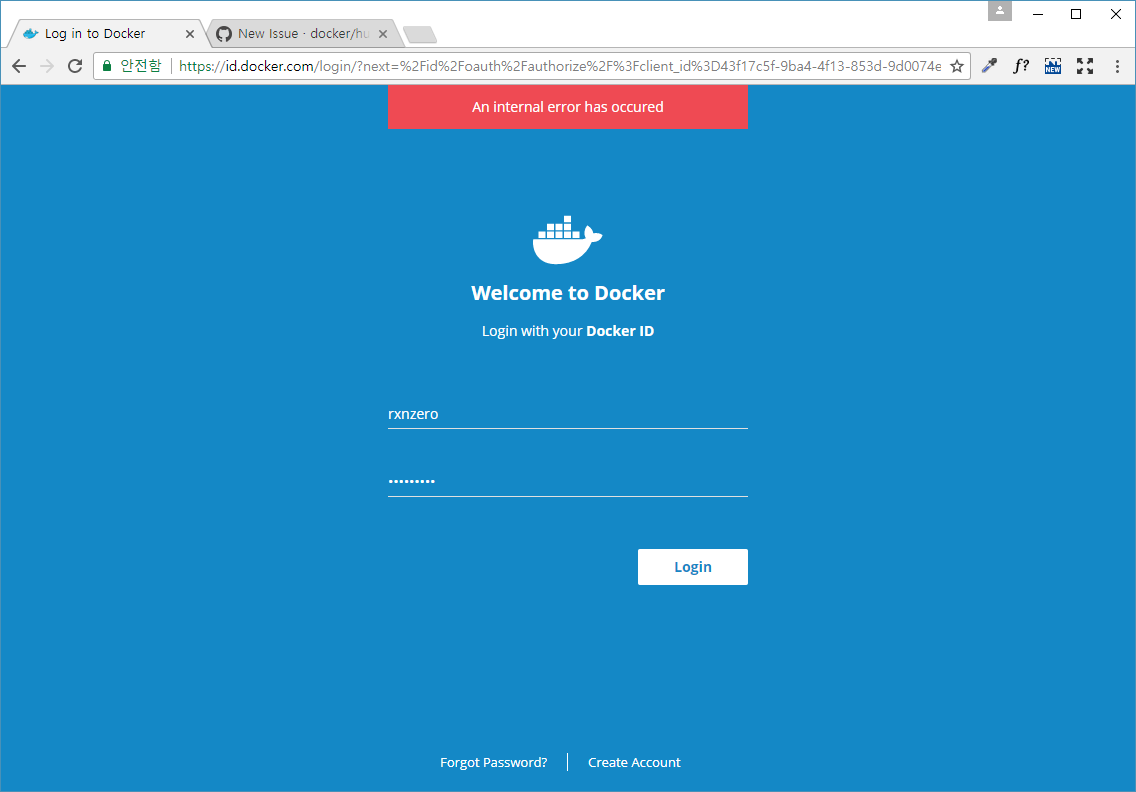
Este servicio no es más que un *playground* para que, quien quiera pueda explorar la creación y gestión de contenedores virtuales basados en Docker. Vamos a ver en qué consiste.

Descripción

Play with Docker

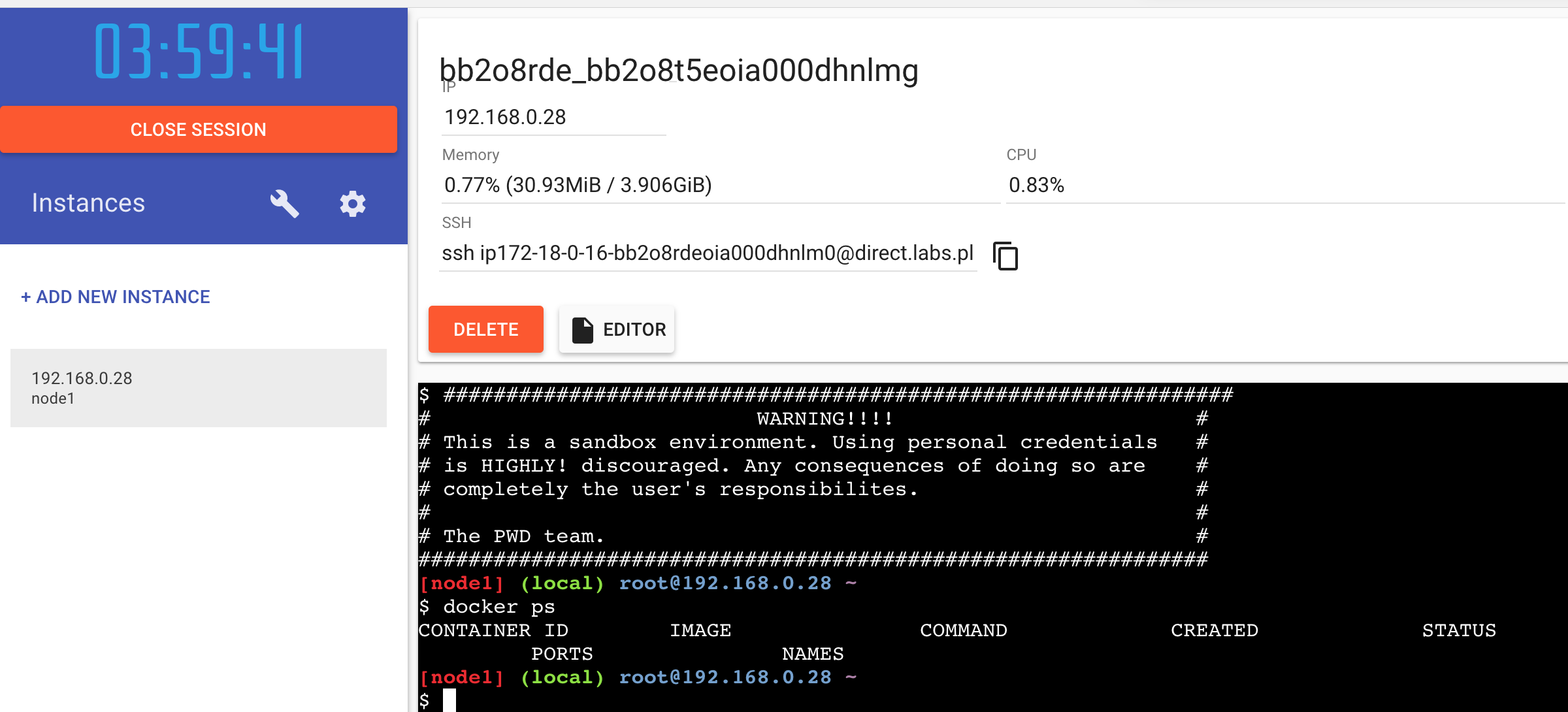
Play with Docker es un servicio *cloud* que nos permite *jugar* con Docker sin necesidad de instalar Docker localmente. Cada vez que accedemos a una sesión en Play with Docker, tenemos **4 maravillosas horas** para experimentar con Docker, hasta que esta se cierra automáticamente. Parece poco tiempo pero es el suficiente para llevar a cabo nuestro humilde y divertido laboratorio. Dicho de otra manera: tenemos acceso casi ilimitado (en recursos) a un super computador orientado a familiarizarnos con contenedores virtuales durante todo el periodo de duración de la clase/laboratorio.

Para acceder a Play with Docker, solo necesitamos una cuenta en Docker. Como en la gran mayoría de servicios, este registro es gratuito, rápido y comporta los típicos pasos de selección de contraseña, envío de correo, confirmación de recepción de correo, etc.



Una vez que ya tenemos acceso a la consola de Play with Docker, podemos solicitar máquinas *reales* (en realidad siguen siendo virtuales, pero a efectos prácticos, las trataremos como si fueran reales) sobre las que ejecutar a su vez contenedores Docker gracias a su *toolkit (*<https://docs.docker.com/engine/reference/commandline/docker/#child-commands>).

Al inicio no habrá ninguna, pero podemos crear la primera (y única que necesitamos) con el comando **Add new instance** que se puede ver en la imagen.



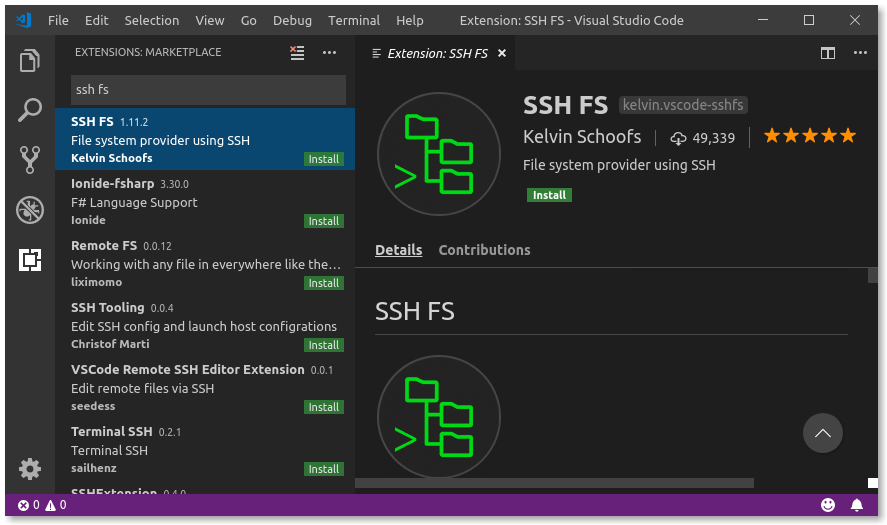
Cuando creamos una instancia, es como si accediéramos a una máquina real (o para lo que a todos los efectos, será una máquina real para nosotros). Se nos abrirá una *sesión de shell* (bash) justo al lado y sobre la que podemos introducir comanos Unix *de toda la vida*, entre ellos los relacionados con Docker. En estas máquinas *cuasi-reales* ya está instalado todo el *stack* de Docker y podemos, a su vez, trabajar con contenedores virtuales basados en esta tecnología de vanguardia.

Fíjate que la instancia que has creado en Play with Docker es como si se tratara de un ordenador tuyo que estuviera en red y en ese ordenador tuvieras instalado Docker. Así de sencillo. A partir de ese momento ya no necesitamos acceder a Play with Docker mediante el navegador (aunque está bien que lo dejes *aparcado* a un lado en caso de necesidad). Usaremos VS Code y acceso SSH tradicional.

**Importante**: tómate nota del usuario que te da Play with Docker y que empieza por ip y acaba en @direct… (Ejemplo: ip172-18-0-27-bg3pvjs3uhdg008ir0fg). Copia esa cadena de texto y guárdatela para futuros usos. A lo largo de este trabajo nos referiremos a ella como usuariopwd.

Software a instalar localmente

Estos son los distintos programas/paquetes que necesitas en tu sistema local:

* Visual Studio Code (VS Code, para los *amigos*).
* Extensiones de VS Code a instalar (no te olvides de darle a reload al instalar cada extensión):
  + SSH FS (<https://github.com/SchoofsKelvin/vscode-sshfs>). **Preguntas**: ¿qué es el sistema de archivos SSHFS, qué relación tiene con el protocolo SSH y por qué es tan revolucionario?, ¿qué es FUSE (usado internamente por SSHFS)? ¿qué otros sistemas de archivos basados en FUSE conoces? Ejemplos:
* <https://github.com/libfuse/libfuse/wiki/Filesystems>
* <https://github.com/koding/awesome-fuse-fs>
  + 
  + VRML V2.0 Editing Kit

(<https://marketplace.visualstudio.com/items?itemName=icetea78.wrl>)

* + Python (<https://marketplace.visualstudio.com/items?itemName=ms-python.python>). Ojo, no es necesario instalar Python ni vamos a trabajar con un intérprete de este lenguaje en nuestra máquina local. Solo queremos esta extensión para que VS Code haga sintaxis de color de código Python (que será ejecutado por un contenedor Docker que crearemos en nuestra instancia *pseudoreal* en el servicio Play with Docker).
* Putty, en el caso de Windows. (<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>)
* Instant Player (<https://www.instantreality.org/downloads/>) para visualizar las trazas de partículas. Se trata de un visor de instrucciones VWRL disponible tanto para macOS, Windows, Ubuntu, Red Hat y SUSE. Para solucionar el problema de que *falta un archivo* MSVCP110.dll en el caso de algunas versiones de Windows, instalad el *runtime* mínimo de *Microsoft Visual C++ 2012 Redistributable* para 64 bit desde aquí (<https://www.microsoft.com/en-au/download/confirmation.aspx?id=30679>). **Preguntas:** ¿qué otros formatos estándar de representación 3D conoces (Collada, Stanford, Wavefront, X3D Extensible, Standard Tessellation Language, etc.)?, ¿quién desarrolla Instant Player y qué contribuciones ha hecho al mundo de la tecnología?
* Paraview (<https://www.paraview.org/>), de Kitware, pero solo se recomienda su uso en el caso de que, por alguna razón Instant Player no funcione de ninguna manera (quizás en el caso de que tu arquitectura sea todavía 32 bit). Para ver las trazas de las partículas correctamente, elige el modo Wireframe (barra de herramientas). **Pregunta:** ¿en qué proyectos está involucrada la empresa Kitware y cómo han cambiado el mundo de la informática, la tecnología, la ciencia y la medicina? Pista: CMake, VTK, ITK, etc.
* Docker… ¡No! No es necesario instalar Docker, pero ya que estamos… **Pregunta:** ¿qué es Docker?, ¿qué son los *contenedores virtuales*?, ¿qué alternativas a Docker existen (OpenVZ, LXC, Vagrant, rkt, etc.)? Resume brevemente la historia de Docker. ¿Por qué Docker se ha convertido en un *software* tan popular? Aquí (<https://www.youtube.com/watch?v=cYsVvV1aVss&t=1066s>) tienes una pista.

Ejecución del laboratorio

Como hemos comentado antes, no vamos a usar la web de Play with Docker. En su lugar, usaremos herramientas de vanguardia y profesionales, como SSH y SSH FS.

* Crea una instancia en Play with Docker si no lo has hecho ya.
* Abre una ventana vacía de VS Code (New Window).
* Muestra el Side Bar (View ➡️ Appearance ➡️ Toggle Side Bar).
* Edita las preferencias de usuario de la extensión SSH FS, indicando que quieres añadir una nueva configuración como se indica en la figura.

![](data:image/png;charset=UTF-8;base64,)

* Saldrá un mensaje pidiéndote un nombre para la conexión. Llámala, por ejemplo, play-with-docker y luego edítala en el fichero de tipo jsonc (<https://komkom.github.io/>)que se te muestra. Configura los valores, prestando atención a tu usuario en Play with Docker y el *path* completo al fichero ppk (del que hablaremos a continuación). Si usas barras invertidas ("") para el caso de Windows, acuérdate de escaparlas ("\"). En versiones relativamente modernas de Windows, puedes usar la barra directa ("/") que ya no es necesario escapar. **Importante**: te aconsejamos que crees una carpeta de trabajo (CarpetaTrabajo) y operes siempre desde ella. A continuación tienes un ejemplo de segmento JSON de configuración de conexión, que finalmente debería aparecer integrado como parte del código JSON en la configuración de VS Code (fichero settings.json). Cuando le des a grabar, seguramente, la extensión SSH FS te preguntará si quieres abrir ya la conexión.

"sshfs.configs": [{

"label": "Play with Docker",

"root": "/root",

"host": "direct.labs.play-with-docker.com",

"port": 22,

"username": "usuariopwd",

"privateKeyPath": "/path/a/CarpetaTrabajo/geant4lab.ppk",

"name": "play-with-docker"

}]

* La ruta al fichero ppk no es más que un fichero de texto con este contenido. Grábalo en la CarpetaTrabajo como geant4lab.ppk (¡no cambies nada!). **Preguntas**: ¿qué es un fichero ppk?, ¿qué es una clave RSA público/privada?

PuTTY-User-Key-File-2: ssh-rsa

Encryption: none

Comment: [pammac@PammacMBP.local](mailto:pammac@PammacMBP.local)

Public-Lines: 6

AAAAB3NzaC1yc2EAAAADAQABAAABAQC2yczxCkLs37pQCiar4BhwjXL5hZC8uOIF  
MimWeBtEGfbK9ZAQx3/LE2kNXdRK1BvgM7caPGtGaPJLSwJ4HpvVSq+Rby/TYjys  
0SCBT5t6h8MD8Lc5otibz/T9FUmp5tUPkOvI5gc6B31fph9esL5+cc4dJU7fqO4k  
wAM07dQI3n7VFJrcgA+BVRYfTZZOhfEsMJN2fm0pvif3W3RRYdf9eor01ft9IAzS  
QleMIkT4Io48oBbYQMRWu9NDe9W6rYMeBTvinL3zv5es1GGXue4Md9KDmojwFbpo  
gMXB0zd7nc5se1bqHe/CeUnBmwuPx3xscdXAeaiBxanMXoDlgC8L  
Private-Lines: 14

AAABAQClt17uSPdT+aHYXtugh5KguBy8y7Lk5llDjK7XlBxkKYXti6QJElVxry9W

xvJoxNRg94HRb2VyeELysBaVAGsXfJafq8YYRr/rjc2CbsHu3JaZQ+tZfaIT4P2j

VWFrf7M6YZDimmmC2rd3uAxItMR/rn3HPsO7UlRVTR+bJONPBaBPLZUyKLVnCVeh

QRLm4g6XjvPgHHfJ9DIp61pTLl7UPGl58gVrylDuzdqFWl6Y4J4TJ6LTlPgKQlWA

mJdyu9SWHCSNVyIpTB/OFnzXif4nrfI3wEzvOhdqxhgqOjcLbIRzCX+gjrYurepd

7CiXDGQTevN5Xs/dTaRUslUOfA/BAAAAgQDf6O9M1RBtManYoan3rIKMhtlFbp9j

0SS06pxnFuAORTavUWz0Gf+QeojoJ5bBRwK6/GewTTCUWA9emQcEVFr4j7+l/PLt

AuaKXFtHJBVEq9OKbOl3fjJdHaeAohQkq3VRGNa7D4rGQJ5+xZ+UqoUJvCfj+rXV

jOpyprBMjO3rxwAAAIEA0Pwl+a9Xv7+p3gA9DpZ6s12fOflS69AnlMS3lo8FhOil

NTIG979vgJyq0Ivr+cSjFkbHfUdeCXZFMCnlUvQlxOcywQuzeDdYEJ0jm6Y/m+Sa

7q+48PounZ/tsobVF/WKn3PqJ8nUhT1zapfJqdOf4SlqoBjk1wGNNJYfokN4up0A

AACBAMU13cdg+DSG4i/Up+XT+ts3W4j1dUW8kWXVEIzwX0jCJLye/SkCMMOYROik

1qZtzKuBxch+BynnbT99ekHqWQzpa/ChrVGzJk4taDwdPr5NkkI4CsxUSImx9CZ7

6I+YwVCualZqleZq/gPtygtfkuJQEkDuDRDB81GFW+V5Ci1U

Private-MAC: d390627f9caec35e756050f47a01d5948e832d65

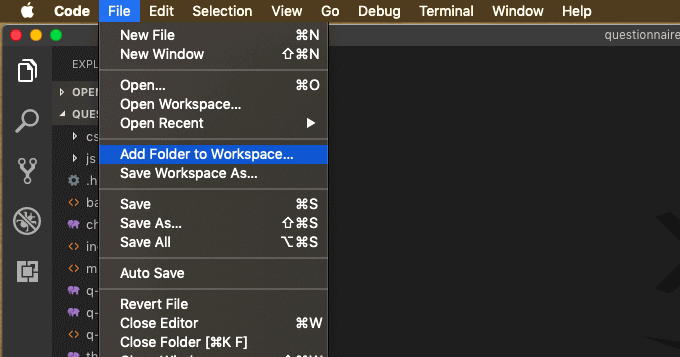
* Si no has abierto todavía la conexión SSH FS a Play with Docker, en la sección ***SSH FILE SYSTEMS*** deberías ver un ítem llamado **Play with Docker** (o el nombre al que hayas puesto en la configuración de la conexión en el campo label). Haz click con el botón derecho y se abrirá un menú desplegable. Selecciona ***Connect as Workspace folder***.

![](data:image/png;charset=UTF-8;base64,)

* Si todo está bien configurado, verás que el símbolo de conexión (⚪) cambia a color verde y que la vista en árbol de *Workspace* se puebla con los contenidos de la carpeta /root de la instancia de Play with Docker. Es como si pudiéramos ver la carpeta de nuestra máquina en red en la propia ventana de VS Code.

¡Ojo! Play with Docker resetea la conexión SSH cada cierto tiempo (aunque el indicador en VS Code esté en verde). Por ello, si ves que no responde el visor de archivos de VS Code (no se actualiza), selecciona la opción ***Reconnect Workspace folder***. De hecho es posible que la propia extensión detecte la caída de la conexión y te muestre un mensaje del tipo *SSH FS Play with Docker disconnected*.

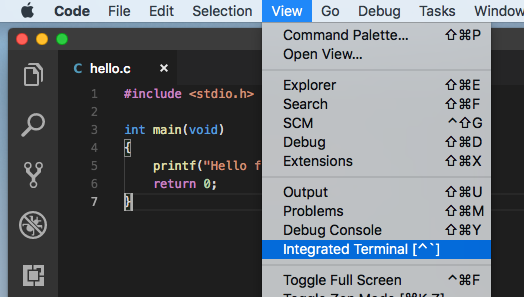
* Para trabajar con comodidad, añade al *Workspace* el directorio de trabajo donde se encuentren el fichero ppk o (File ➡️ Add folder to Workspace). En este directorio será también donde se guarden los resultados de las simulaciones (ficheros .wrl que luego comentaremos).



Descarga de la imagen Docker del laboratorio

Lo primero que tenemos que hacer es descargarnos la imagen de la pila de contenedores que conforman una imagen con todo lo necesario para ejecutar Geant4. Esta imagen se encuentra ya en el Docker *Hub* (<https://hub.docker.com/>). **Preguntas**: ¿qué es el *Hub* de Docker?, ¿cuál es la diferencia entre una imagen y un contenedor?

Para enviar esta orden ejecutaremos un comando remoto por SSH. Estos comandos remotos los invocaremos directamente desde el terminal integrado de VS Code. Nunca salimos de VS Code si no es para visualizar el resultado de las simulaciones.



* En el caso de macOS, primero graba el contenido de esta cadena de texto en un fichero llamado geant4lab.key en la CarpetaTrabajo y dale permisos como se indica aquí (<https://stackoverflow.com/questions/9270734/ssh-permissions-are-too-open-error>). **Pregunta**: ¿qué representa este fichero y cómo se relaciona con un ppk?

-----BEGIN OPENSSH PRIVATE KEY-----

b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABFwAAAAdzc2gtcn

NhAAAAAwEAAQAAAQEAtsnM8QpC7N+6UAomq+AYcI1y+YWQvLjiBTIplngbRBn2yvWQEMd/

yxNpDV3UStQb4DO3GjxrRmjyS0sCeB6b1UqvkW8v02I8rNEggU+beofDA/C3OaLYm8/0/R

VJqebVD5DryOYHOgd9X6YfXrC+fnHOHSVO36juJMADNO3UCN5+1RSa3IAPgVUWH02WToXx

LDCTdn5tKb4n91t0UWHX/XqK9NX7fSAM0kJXjCJE+CKOPKAW2EDEVrvTQ3vVuq2DHgU74p

y987+XrNRhl7nuDHfSg5qI8BW6aIDFwdM3e53ObHtW6h3vwnlJwZsLj8d8bHHVwHmogcWp

zF6A5YAvCwAAA9D1Nbub9TW7mwAAAAdzc2gtcnNhAAABAQC2yczxCkLs37pQCiar4BhwjX

L5hZC8uOIFMimWeBtEGfbK9ZAQx3/LE2kNXdRK1BvgM7caPGtGaPJLSwJ4HpvVSq+Rby/T

Yjys0SCBT5t6h8MD8Lc5otibz/T9FUmp5tUPkOvI5gc6B31fph9esL5+cc4dJU7fqO4kwA

M07dQI3n7VFJrcgA+BVRYfTZZOhfEsMJN2fm0pvif3W3RRYdf9eor01ft9IAzSQleMIkT4

Io48oBbYQMRWu9NDe9W6rYMeBTvinL3zv5es1GGXue4Md9KDmojwFbpogMXB0zd7nc5se1

bqHe/CeUnBmwuPx3xscdXAeaiBxanMXoDlgC8LAAAAAwEAAQAAAQEApbde7kj3U/mh2F7b

oIeSoLgcvMuy5OZZQ4yu15QcZCmF7YukCRJVca8vVsbyaMTUYPeB0W9lcnhC8rAWlQBrF3

yWn6vGGEa/643Ngm7B7tyWmUPrWX2iE+D9o1Vha3+zOmGQ4pppgtq3d7gMSLTEf659xz7D

u1JUVU0fmyTjTwWgTy2VMii1ZwlXoUES5uIOl47z4Bx3yfQyKetaUy5e1DxpefIFa8pQ7s

3ahVpemOCeEyei05T4CkJVgJiXcrvUlhwkjVciKUwfzhZ814n+J63yN8BM7zoXasYYKjo3

C2yEcwl/oI62Lq3qXewolwxkE3rzeV7P3U2kVLJVDnwPwQAAAIEAxTXdx2D4NIbiL9Sn5d

P62zdbiPV1RbyRZdUQjPBfSMIkvJ79KQIww5hE6KTWpm3Mq4HFyH4HKedtP316QepZDOlr

8KGtUbMmTi1oPB0+vk2SQjgKzFRIibH0Jnvoj5jBUK5qVmqV5mr+A+3KC1+S4lASQO4NEM

HzUYVb5XkKLVQAAACBAN/o70zVEG0xqdihqfesgoyG2UVun2PRJLTqnGcW4A5FNq9RbPQZ

/5B6iOgnlsFHArr8Z7BNMJRYD16ZBwRUWviPv6X88u0C5opcW0ckFUSr04ps6Xd+Ml0dp4

CiFCSrdVEY1rsPisZAnn7Fn5SqhQm8J+P6tdWM6nKmsEyM7evHAAAAgQDQ/CX5r1e/v6ne

AD0OlnqzXZ85+VLr0CeUxLeWjwWE6KU1Mgb3v2+AnKrQi+v5xKMWRsd9R14JdkUwKeVS9C

XE5zLBC7N4N1gQnSObpj+b5Jrur7jw+i6dn+2yhtUX9Yqfc+onydSFPXNql8mp05/hKWqg

GOTXAY00lh+iQ3i6nQAAABZwYW1tYWNAUGFtbWFjTUJQLmxvY2FsAQID

-----END OPENSSH PRIVATE KEY-----

* Ahora ya podemos enviar la orden mediante conexión SSH, que en el caso de macOS será:

ssh usuariopwd@direct.labs.play-with-docker.com -i geant4lab.key "/usr/local/bin/docker pull pammacdotnet/geant4lab"

* En el caso de Windows, asegúrate de añadir previamente el directorio donde ha quedado Putty instalado (donde se encuentren los programas putty.exe y plink.exe) a la variable path de la configuración de VS Code:

"terminal.integrated.env.windows": {

"path": "%path%;C:/path/a/carpeta/de/plink-y-putty"

}

* Ahora ya puedes ejecutar el mismo comando desde el terminal integrado de VS Code para Windows.

plink.exe -ssh -l usuariopwd direct.labs.play-with-docker.com -i geant4lab.ppk "/usr/local/bin/docker pull

pammacdotnet/geant4lab"

* Este comando Docker lo que hace es descargarse la imagen geant4lab del usuario pammacdotnet en el *Hub* de Docker. La descarga se produce desde este *Hub* al disco de la instancia en Play with Docker.
* Crea un fichero simulacion.py en el *Workspace* remoto (que has accedido/montado por SSH FS) y puéblalo con el código siguiente. Si VS Code te pide que selecciones un intérprete Python, puedes ignorarlo, ya que no vamos a trabajar con un Python local, sino con un intérprete remoto que, a su vez, reside en un contenedor Docker. Cuando grabes este fichero, se grabará en el *disco duro remoto*.

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

import Geant4

from Geant4 import \*

import g4py.NISTmaterials

import g4py.ezgeom

from g4py.ezgeom import G4EzVolume

import g4py.EMSTDpl

import g4py.ParticleGun

import g4py.MedicalBeam

import os.path

# Tipo de render: "surface" o "wireframe"

tipo\_de\_render = "wireframe"

# Tipo de partículas: gamma, e+ ó e-

tipo\_de\_particulas = "e-"

# Número de partículas en el haz

numero\_de\_particulas = 20

# Energía de las particulas (en MeV)

energia\_de\_las\_particulas = 100

# Grosor del fantoma (en cm)

grosor\_del\_fantoma = 20

# Material del fantoma

# La lista de materiales la tienes en el enunciado

# http://geant4.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/BackupVersions/V9.4/html/apas09.html

# Aquí tienes algunos: G4\_Al, G4\_Si, G4\_Ar, G4\_Cu, G4\_Fe, G4\_Ge, G4\_Ag, G4\_W, G4\_Au, G4\_Pb, G4\_AIR, G4\_Galactic, G4\_WATER H\_2O, G4\_CESIUM\_IODIDE, G4\_SODIUM\_IODIDE, G4\_PLASTIC\_SC\_VINYLTOLUENE, G4\_MYLAR

material\_del\_fantoma = "G4\_WATER"

# Colores de las trazas para que las puedas seguir en el diagrama 3D

color\_electrones = "yellow"

color\_positrones = "green"

color\_fotones = "white"

color\_protones = "blue"

rand\_engine = Ranlux64Engine()

HepRandom.setTheEngine(rand\_engine)

HepRandom.setTheSeed(20050830L)

g4py.NISTmaterials.Construct()

g4py.ezgeom.Construct()

g4py.EMSTDpl.Construct()

g4py.ParticleGun.Construct()

# El espacio hasta llegar al fantoma, supongamos que es aire, pero también

# puedes simular que es el vacío intergaláctico con "G4\_Galactic"

air = G4Material.GetMaterial("G4\_AIR")

g4py.ezgeom.SetWorldMaterial(air)

g4py.ezgeom.ResizeWorld(120.\*cm, 120.\*cm, 400.\*cm)

# Definimos un fantoma y lo situamos en la escena

phantom\_material = G4Material.GetMaterial(material\_del\_fantoma)

phantom = G4EzVolume("PhantomBox")

phantom\_zwidth = grosor\_del\_fantoma\*cm

phantom\_zlocation = 100.\*cm

phantom.CreateBoxVolume(phantom\_material, 100.0 \* cm, 100.0 \* cm, phantom\_zwidth)

phantom.SetColor(0., 0.9, 1.0)

phantom\_box\_pv = phantom.PlaceIt(G4ThreeVector(0.\*cm, 0.\*cm, phantom\_zlocation))

# Creamos un haz de partículas y lo dirigimos contra el fantoma

beam = g4py.MedicalBeam.Construct()

beam.particle = tipo\_de\_particulas

beam.kineticE = energia\_de\_las\_particulas\*MeV

beam.sourcePosition = G4ThreeVector(0.\*cm, 0.\*cm, -90.\*cm)

beam.fieldXY = [120.\*cm, 120.\*cm]

beam.SSD = 190.\*cm

# http://geant4.slac.stanford.edu/Presentations/vis/G4VisCommands.pdf

# http://geant4.slac.stanford.edu/Presentations/vis/G4VisAdvanced.pdf

# Los siguientes comandos son para generar la escena en un fichero VRML

gApplyUICommand("/run/initialize")

gApplyUICommand("/vis/viewer/flush")

gApplyUICommand("/vis/open VRML2FILE")

gApplyUICommand("/vis/viewer/refresh")

gApplyUICommand("/vis/scene/create")

gApplyUICommand("/vis/scene/add/volume")

gApplyUICommand("/vis/drawVolume")

gApplyUICommand("/vis/modeling/trajectories/create/drawByParticleID")

gApplyUICommand("/vis/modeling/trajectories/drawByParticleID-0/set gamma " + color\_fotones)

gApplyUICommand("/vis/modeling/trajectories/drawByParticleID-0/set proton " + color\_protones)

gApplyUICommand("/vis/modeling/trajectories/drawByParticleID-0/set e- " + color\_electrones)

gApplyUICommand("/vis/modeling/trajectories/drawByParticleID-0/set e+ " + color\_positrones)

gApplyUICommand("/vis/sceneHandler/attach")

gApplyUICommand("/vis/viewer/set/style " + tipo\_de\_render)

gApplyUICommand("/vis/viewer/set/viewpointThetaPhi 70. 10.")

gApplyUICommand("/vis/viewer/zoom 1.")

gApplyUICommand("/tracking/storeTrajectory 1")

gApplyUICommand("/vis/scene/add/trajectories")

gApplyUICommand("/vis/scene/add/hits")

gApplyUICommand("/vis/scene/add/trajectories smooth")

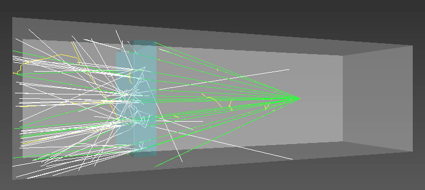
gApplyUICommand("/vis/scene/endOfEventAction accumulate")

gRunManager.Initialize()

gRunManager.BeamOn(numero\_de\_particulas)

os.rename("g4\_00.wrl", "simulacion.wrl")

Este programa describe un experimento de física de partículas cuya representación es más o menos esta:



Se trata de un *universo* en forma de *cuboide* donde las partículas subatómicas son aceleradas (por los medios que sean) y son proyectadas con forma *cónica* contra un objetivo (un *target* o *fantoma*) de un material que podemos variar a nuestro antojo en composición, tamaño y posición. Nosotros no nos preocupamos sobre cómo esas partículas han cobrado esa energía, sino solamente por cómo interaccionan con la materia.

**Preguntas**: comenta las líneas más importantes del código anterior. Intenta extraer su significado por sus palabras. Si tienes dudas, pregunta en el foro. ¿En qué unidades viene descrita la energía de las partículas? Verifica que verdaderamente se corresponden con unidades de energía y expresa un par de ejemplos en Julios (unidad de energía en el S.I.).

Ejecución de la simulación

Ahora solo tenemos que lanzar el comando de ejecución de la simulación desde el terminal integrado de VS Code:

* En el caso de macOS:

ssh usuariopwd@direct.labs.play-with-docker.com -i geant4lab.key "/usr/local/bin/docker run -v /root:/root/Geant4Lab pammacdotnet/geant4lab"

* En el caso de Windows:

plink -ssh -l usuariopwd direct.labs.play-with-docker.com -i geant4lab.ppk "/usr/local/bin/docker run -v /root:/root/Geant4Lab pammacdotnet/geant4lab"

Deberías ver que aparece un nuevo fichero en el *Workspace*: simulacion.wrl. Este fichero es un diagrama 3D que puedes ver. Si es necesario pulsa el icono de recarga (↩️) en el *Workspace* o como hemos comentado antes, *reconecta* con la instancia de Play with Docker (botón derecho en la sección de SSH FILE SYSTEMS). Este comando anterior crea un contenedor Docker que tiene *mapeada* la carpeta interna */root/Geant4Lab* con el directorio home del usuario *root* en la instancia Play with Docker. El contenedor tiene como cometido ejecutar un único proceso (que no es otro que un intérprete Python que interpreta las instrucciones de simulacion.py). Como las dos carpetas están enlazadas tanto para lectura como escritura, el fichero simulacion.wrl estará disponible en /root de la instancia Play with Docker.

También puedes comprobar en cualquier momento que se ha generado correctamente ese fichero accediendo mediante la web de Play with Docker o, nuevamente, mediante consola/SSH interactiva a la máquina de Play with Docker. En ambos casos, el comando ls te listará los ficheros de la carpeta *home* (/root) del usuario por defecto (root)

* En el caso de macOS:

ssh usuariopwd@direct.labs.play-with-docker.com -i geant4lab.key

* En el caso de Windows:

putty.exe -ssh -l usuariopwd direct.labs.play-with-docker.com -i geant4lab.ppk

Visualización de la simulación

Una vez que ya tenemos la simulación, solo tenemos que visualizarla con Instant Player (o Paraview). Para ello, tenemos que *traer* (o mejor expresado, *descargar*). Lo podemos lograr gracias a lo comandos scp y pscp:

En el caso de macOS:

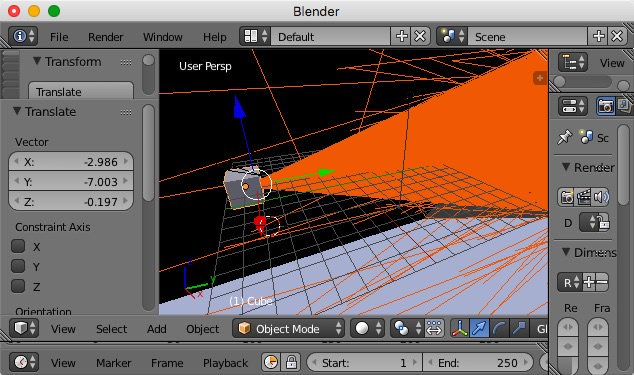
scp -i geant4lab.key usuariopwd@direct.labs.play-with-docker.com:/root/simulacion.wrl simulacion.wrl

En el caso de Windows, usamos pscp.exe, que viene con el paquete Putty.

pscp.exe -i geant4lab.ppk usuariopwd@direct.labs.play-with-docker.com:/root/simulacion.wrl simulacion.wrl

Una vez descargada, abrimos el fichero con Instant Player o, si ya está el fichero abierto, puedes simplemente recargarlo (File ➡️ Reload). Lo que vemos en el visualizador son trazas de partículas, es decir, solo vemos partículas que estén en movimiento. Las partículas o materia *paradas*, no se muestran.

Por último: también sería posible visualizar los ficheros wrl con Blender <https://www.blender.org/>:



Entregable del laboratorio

Se pide un informe técnico personal, elegante y bien presentado de todas las tareas necesarias para la ejecución de la actividad. Es decir, reproducid en vuestro escenario todos los pasos realizados y que más o menos imiten a los que ya vienen descritos en este documento (instalación de *software*, configuración, accesos remotos, etc.).

Contestad a todas las preguntas intermedias que se proponen.

Buscad e identificad todos los tipos de interacciones de partículas con la materia posibles, modificando para ello todos los parámetros de la simulación. Comenta claramente el *setup* experimental definido en cada caso (energía de las partículas, si se trata de fotones, electrones o positrones, posición del haz, tipo de fantoma, material del mismo, etc.). Ejemplifica cada interacción con una captura de pantalla bien explicada del fenómeno físico que se pueda apreciar. A continuación tienes un resumen de las interacciones más básicas a estas energías.

El documento final debe tener formato PDF o HTML *inline* (todos los recursos incrustados en **único fichero**). Para generar el HTML *inline* puedes exportar primero a HTML desde Typora (esto creará un fichero HTML y un subdirectorio con todas las imágenes que hayas incluido en tu fichero original *Markdown*) y luego condensar todos los recursos en un único documento. Para llevar a cabo esta labor de *condensación* tienes a tu disposición varias herramientas basadas en Node.js/npm. La mejor es *inline assets* <https://www.npmjs.com/package/inline-assets>, con la que podrás fusionar todos los recursos de una web en un único documento HTML de la siguiente manera:

inline-assets laboratorio\_geant4.html

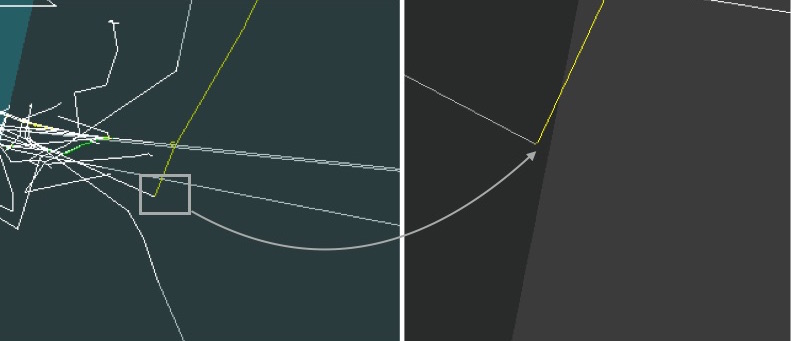
laboratorio\_geant4\_inline.html

Anexo: interacciones de partículas con la materia

Vamos a describir las más habituales.

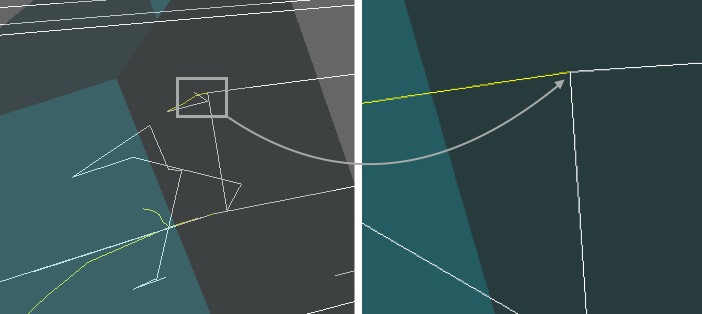
Efecto fotoeléctrico

El efecto fotoeléctrico se produce cuando tiene lugar una interacción entre un fotón (partícula incidente) y un átomo. La consecuencia de una interacción fotoeléctrica es la emisión de electrones (fotoelectrones), debido a la absorción total de la energía del fotón por el electrón ligado. El fotón es absorbido completamente y *desaparece* de la escena.



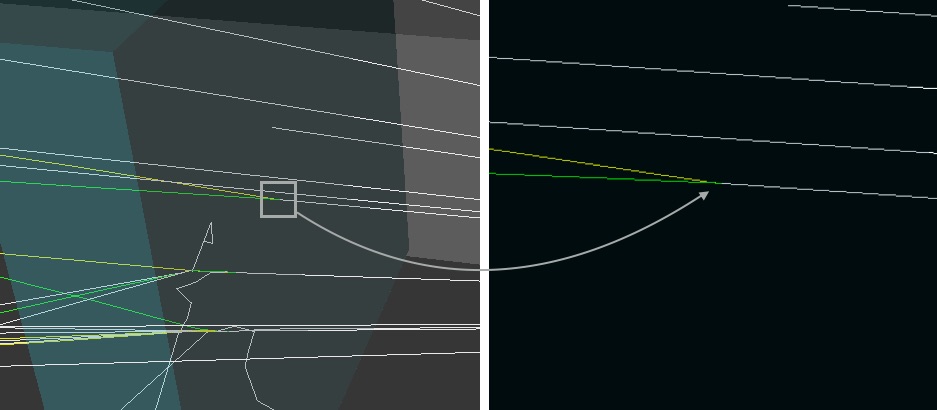
Efecto Compton

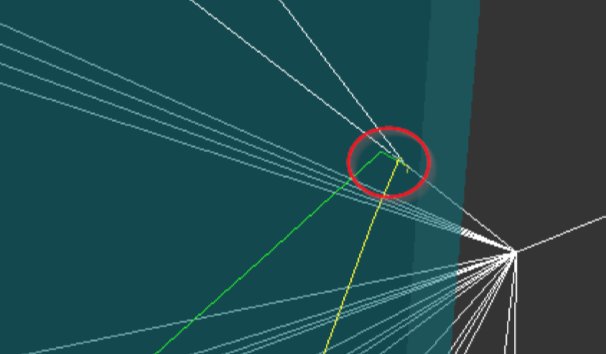
El efecto Compton tiene lugar en la interacción de un fotón (partícula incidente) y un electrón poco ligado al átomo. Se produce entonces una colisión *casi elástica*. Al colisionar el fotón con el electrón este es desviado, y el fotón original también. Puede asemejarse a un *choque de bolas de billar*. A veces si la energía del fotón incidente es muy justa, el electrón (o electrones, en el caso de una colisión múltiple) desviado apenas avanza (su traza es muy pequeña, casi imperceptible) y da la sensación de que el fotón (que, recordamos, no es más que luz) está *zigzagueando*. **Pregunta**: ¿cuánto tiempo puede estar *serpenteando* un fotón que *nace* en el centro del Sol hasta llegar a la retina de cualquiera de tus ojos?



Producción de pares

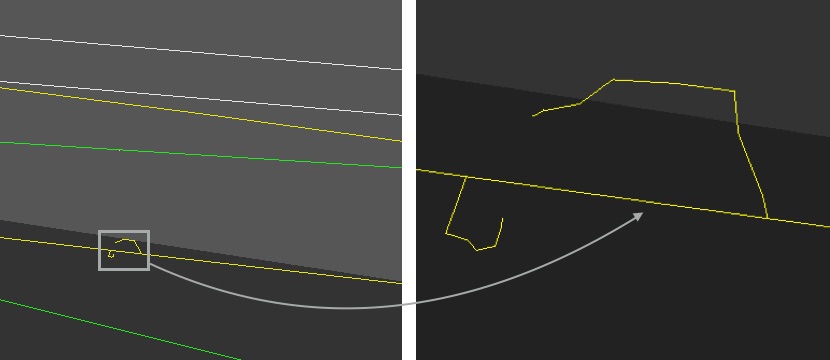
El efecto de creación de pares tiene lugar con fotones de alta energía (> 1.1 MeV), y representa un **proceso de materialización de energía** en el sentido de la mecánica relativista. El fenómeno que tiene lugar es la desaparición del fotón en el campo del núcleo, y la creación en su lugar de un par positrón-electrón (materia y antimateria). **Pregunta**: ¿qué es un positrón?, ¿quién postuló su existencia teórica y cuándo y cómo se descubrió experimentalmente?





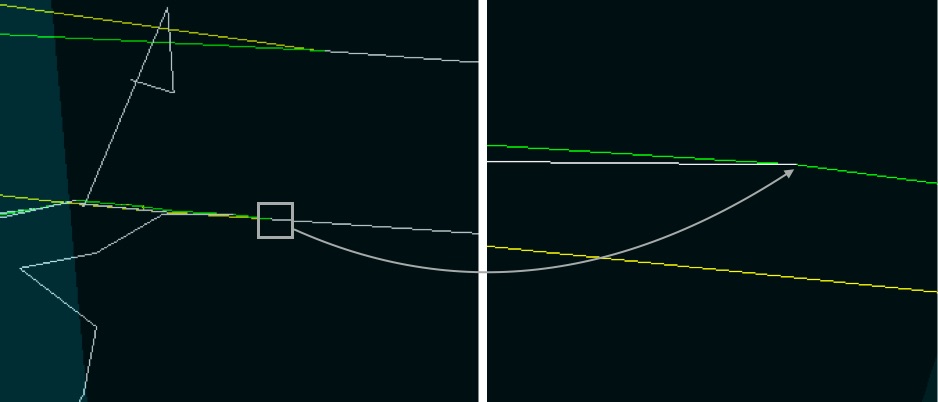
Ionización

Si en los choques de una partícula cargada con los electrones atómicos, la energía transferida es superior a la energía de enlace del electrón que *recibe el impacto*, este abandona el átomo y en consecuencia se crea un ion positivo. El electrón expulsado puede a continuación zigzaguear hasta ser capturado por un átomo del entorno (la traza deja de verse porque, como hemos comentado, el electrón se *para*).



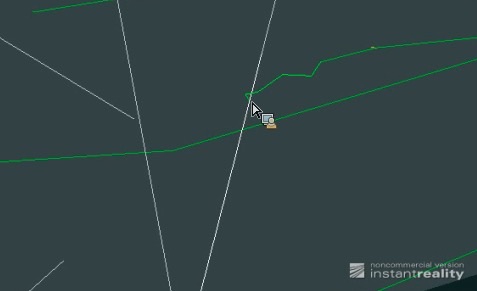
Bremsstrahlung

Cuando una partícula incidente con masa y carga eléctrica interacciona con un núcleo atómico, puede sufrir una variación súbita de su velocidad, emite radiación electromagnética (un fotón).



Aniquilación de pares

Si un positrón y un electrón se encuentran, se *aniquilan* creando dos fotones que son emitidos en direcciones opuestas. **Pregunta**: ¿qué relación tiene la modalidad radiológica PET con la aniquilación de pares electrón-positrón?, ¿cómo se *trasladan* positrones hasta el interior de las células cancerígenas?



Rúbrica

* Se valorarán ejemplos gráficos de cada interacción propuesta en el anexo, (peso 40%).
* Se valorará una correcta descripción de la implementación técnica, (peso 30%).
* Se responden a todas las cuestiones planteadas a lo largo del enunciado, (peso 30%).

Extensión máxima de la actividad: 5 páginas como mínimo.