

Estructura de los directorios en dune-pdelab-tutorials

C++ Review Dune

```
tutorial-<usuario>
├── BSD
├── c++
│   ├── CMakeLists.txt
│   ├── doc
│   ├── exercise
│   └── slides
├── CC-BY-SA
├── cmake
│   └── modules
├── CMakeLists.txt
├── config.h.cmake
├── COPYING
├── dune.module
├── dune-pdelab-tutorials.pc.in
├── gridinterface
│   ├── CMakeLists.txt
│   ├── exercise
│   └── slides
├── latexstyle
│   └── exercise.sty
├── LICENSE.md
├── overview
│   ├── abstractions.bib
│   ├── abstractions.tex
│   ├── CMakeLists.txt
│   ├── debug.opts
│   ├── exercise.sty
│   ├── exercise-workflow.tex
│   ├── overview.tex
│   └── release.opts
├── README.md
├── stamp-vc
├── tutorial00
│   ├── CMakeLists.txt
│   ├── doc
│   ├── exercise
│   ├── slides
│   └── src
├── tutorial01
│   ├── CMakeLists.txt
│   ├── doc
│   ├── exercise
│   ├── slides
│   └── src
├── tutorial02
│   ├── CMakeLists.txt
│   ├── doc
│   ├── slides
│   └── src
├── tutorial03
│   ├── CMakeLists.txt
│   ├── doc
│   ├── exercise
│   ├── slides
│   └── src
├── tutorial04
│   ├── CMakeLists.txt
│   ├── doc
│   ├── exercise
│   ├── slides
│   └── src
├── tutorial05
│   ├── CMakeLists.txt
│   ├── doc
│   ├── exercise
│   ├── slides
│   └── src
├── tutorial06
│   ├── CMakeLists.txt
│   ├── doc
│   ├── exercise
│   ├── slides
│   └── src
├── tutorial07
│   ├── CMakeLists.txt
│   ├── doc
│   ├── exercise
│   ├── slides
│   └── src
├── tutorial08
│   ├── CMakeLists.txt
│   ├── doc
│   └── src
├── tutorial09
│   ├── CMakeLists.txt
│   ├── exercise
│   ├── slides
│   └── src
├── TUTORIALLIST
├── workflow
│   ├── CMakeLists.txt
│   ├── exercise
│   └── slides
```

Los programas del repositorio `dune-pdelab-tutorials` se ejecutarán en Gitpod, una aplicación de Kubernetes de código abierto para entornos de desarrollo automatizados y listos para codificar fuera de la caja, que nos otorga 50 horas gratuitas de uso durante cada mes. El hipervisor KVM que emplea emula un sistema operativo basado en Arch Linux con el módulo `dune-pdelab` más algunas herramientas instaladas. En este entorno, tendrá un directorio `/workspace/tutorial-<usuario>`, que tiene la estructura mostrada a la izquierda. Una vista de árbol de este se muestra en el panel izquierdo, la lista completa de ejercicios está en el archivo `TUTORIALLIST`.

DUNE Numerics utiliza `cmake` como sistema de construcción. En `cmake`, hay una clara separación entre el *directorio fuente*, que generalmente está versionado (aquí: el subdirectorio `tutorial-<usuario>`) y el *directorio de compilación*, donde se encuentran los ejecutables, documentación generado por Lua \LaTeX , archivos `vtk`, `gnuplot`, etc. El directorio de compilación será `/workspace/build`.

El directorio `/workspace/build` refleja la estructura del directorio fuente. Debería navegar por defecto al directorio “build” del ejercicio actual y trabajar allí. Cada directorio de este módulo corresponde a un tutorial (`tutorial0[0-9]`). Todos comparten la siguiente estructura: el directorio `src` contiene el código de ejemplo que se mostró en la conferencia. El directorio `doc` contiene las fuentes de \LaTeX de una explicación detallada del tutorial. El directorio `exercise`, que es relevante para este curso, se subdivide aún más: `task` contiene el código esqueleto sobre el que trabajar durante el ejercicio, `doc` contiene la fuentes de la hoja de ejercicios y `solution` contiene lo que esperas obtener.

En caso de que no esté familiarizado con un sistema GNU/Linux, debajo encontrará una pequeña lista de comandos frecuentes para el desarrollo de los ejercicios:

- `sudo pacman -Syu --noconfirm` actualiza la base de datos de los repositorios e instala paquetes actualizados.
- `man <command>` muestra las páginas del manual del comando, en caso de estar disponible, digite `q` para salir. También es válido `<command> --help`.
- `tldr <command>` muestra las páginas del manual de la comunidad.
- `mkdir -p <name>` crea una carpeta `<name>` sin contenido en el espacio de trabajo actual.
- `cmake -S /workspace/tutorial-<usuario> -B /workspace/build` crea una carpeta `build` sin contenido en el espacio de trabajo actual. En general, `cmake -S ruta/al/directorio/padre/a/CMakeLists.txt`.
- `cd <dir>` cambia el directorio de trabajo actual a `dir` (y al directorio del usuario, si se omite la ruta).
- `ls` lista el contenido del directorio de trabajo actual.
- `pwd` imprime el directorio de trabajo.
- `g++ <options> <sources>` compila fuentes de C++ (por ejemplo en el primer ejercicio de C++).
- `make <executablename>` (re)compila ejecutables en el directorio de compilación actual. Si se omite el nombre del ejecutable, se compilan todos los ejecutables del directorio actual.
- `paraview` es un programa de visualización para archivos de extensión `vtk` o `vtu`.

Recursos

Sección de tutoriales

- <https://wiki.archlinux.org/title/Bash>
- <https://wiki.archlinux.org/title/Classroom>
- https://wiki.archlinux.org/title/CMake_package_guidelines

Visores DICOM y renderizado de volumen

- https://wiki.archlinux.org/title/List_of_applications/Science#DICOM_viewers_and_volume_rendering

C++ Review dune

- <https://cpp-review-dune.github.io>

CMake

- <https://hsf-training.github.io/hsf-training-cmake-webpage>
- <https://enccs.github.io/cmake-workshop>
- <https://dune-project.org/sphinx/content/sphinx/core/index.html>

Créditos a preCICE#264

- <https://merely-useful.tech/py-rse>
- <https://missing.csail.mit.edu>
- <https://simulation-software-engineering.github.io/homepage>

Pasos en Gitpod

1. Inicie sesión en GitHub y en Gitpod. Ajuste los permisos de escritura en <https://gitpod.io/integrations>.
2. Acepte la invitación del GitHub Classroom https://classroom.github.com/a/uciSBZ_m.
3. Ingrese a la liga del repositorio generado <https://github.com/cpp-review-dune/tutorial-<usuario>>.
4. Acceda a gitpod.io/#https://github.com/cpp-review-dune/tutorial-<usuario>.

¹ Adaptación de *Structure of the course material for the IWR Dune Course, March 2021* .