

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

дисциплина: Архитектура компьютера

Студент: Армихос Гонзалез Карла

Группа: НКАбд-02-24

МОСКВА

2024 г.

Содержание

1. Цель работы

Целью работы является изучить идеологию и применение средств контроля версий.

Приобрести практические навыки по работе с системой git.

2. Задание

3. Теоретическое задание

Системы контроля версий (VCS) — это инструменты, предназначенные для управления изменениями в файлах и документах. Они позволяют отслеживать изменения в коде, координировать работу нескольких разработчиков, а также обеспечивают возможность восстановления предыдущих версий файлов. VCS особенно полезны в программной разработке, где требуется постоянное обновление и модификация кода.

Основные понятия, связанные с VCS, включают хранилище, `commit`, историю и рабочую копию. *Хранилище* — это место, где сохраняются все версии проекта. **Commit** — это зафиксированное изменение в коде, которое сохраняет состояние файлов в хранилище. **История** — это последовательность всех коммитов, которая позволяет отслеживать изменения и возвращаться к предыдущим версиям. **Рабочая копия** — это локальная версия проекта, с которой разработчик работает, внося изменения перед их фиксацией в хранилище.

Системы контроля версий можно разделить на централизованные и децентрализованные. В централизованных VCS, таких как SVN (Subversion), все изменения хранятся на центральном сервере, и разработчики работают с единственной версией проекта. В децентрализованных VCS, таких как Git, каждый разработчик имеет свою полную копию репозитория, что позволяет работать автономно и синхронизировать изменения позже.

При единоличной работе с хранилищем VCS разработчик вносит изменения в рабочую копию, выполняет команду `commit`, чтобы сохранить эти изменения в хранилище, и, если необходимо, просматривает историю изменений для отслеживания прогресса или возврата к предыдущим версиям.

При работе с общим хранилищем VCS порядок действий включает клонирование репозитория, создание веток для новых функций или исправлений, внесение изменений в рабочую копию, выполнение коммитов и отправку изменений в центральное хранилище. Также важна синхронизация изменений с другими разработчиками, чтобы избежать конфликтов.

Основные задачи, решаемые инструментальным средством Git, включают управление версиями, совместная работа над проектами, ведение истории изменений и возможность отката к предыдущим состояниям кода. Git также поддерживает ветвление и слияние, что позволяет параллельно разрабатывать новые функции.

Некоторые основные команды Git включают:

- `git init`: создание нового репозитория.
- `git clone`: копирование удаленного репозитория на локальную машину.
- `git add`: добавление изменений в индекс перед коммитом.
- `git commit`: фиксация изменений в хранилище.
- `git push`: отправка локальных изменений в удалённый репозиторий.
- `git pull`: получение изменений из удалённого репозитория.

- `git branch`: управление ветками.

Примеры использования Git при работе с локальными и удалёнными репозиториями включают:

- Для локального репозитория: `git init`, затем `git add .` и `git commit -m "Первый коммит"` для сохранения изменений.

- Для удалённого репозитория: `git clone <URL>`, чтобы скопировать проект, после чего можно вносить изменения, а затем использовать `git push` для отправки своих коммитов на сервер.

4. Порядок выполнения лабораторной работы

4.1 Базовая настройка git

empezamos haciendo la configuracion en git

```
gkarmikhos@fedora:~$ git config --global user.name "<ArmijosKarla>"
gkarmikhos@fedora:~$ git config --global user.email "karla.armijos2003@gmail.com"
```

рис. 4.1.1

Настроим utf-8 в выводе сообщений git:

```
gkarmikhos@fedora:~$ git config --global user.email "karla.armijos2003@gmail.com"
gkarmikhos@fedora:~$ git config --global core.quotePath false
```

рис. 4.1.2

Зададим имя начальной ветки, Параметр autocrlf, Параметр safecrlf

```
gkarmikhos@fedora:~$ git config --global init.defaultBranch master
gkarmikhos@fedora:~$ git config global core.autocrlf input
fatal: not in a git directory
gkarmikhos@fedora:~$ git config --global core.autocrlf input
gkarmikhos@fedora:~$ git config --global core.safecrlf warn
```

рис. 4.1.3

4.2 Создание SSH ключа

creamos la clave

```
gkarmikhos@fedora:~$ ssh-keygen -C "Karla Armijos <1032244775@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/gkarmikhos/.ssh/id_ed25519):
Created directory '/home/gkarmikhos/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gkarmikhos/.ssh/id_ed25519
Your public key has been saved in /home/gkarmikhos/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:ym9KeZ/83+369EeLjphNgR0a+sKyAS9d0bLCwaBAQeo Karla Armijos <1032244775@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|.+.|
|o|
|o. .|
|+ o o = .|
|.E + . =So o|
| . =.oo . .|
| + *o.. . .o|
| o.+ooo=... ++|
| ..o.o++ooo+B|
+----[SHA256]-----+
gkarmikhos@fedora:~$ cat /home/gkarmikhos/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIO2XGg87A4XtzZfZSp0uMk2yuGZepk3UkeApi83Dv7E3 Karla Armijos <1032244775@pfur.ru>
gkarmikhos@fedora:~$
```

рис. 4.2.1

Copiamos la clave que creamos para crear una nueva clave en SSH

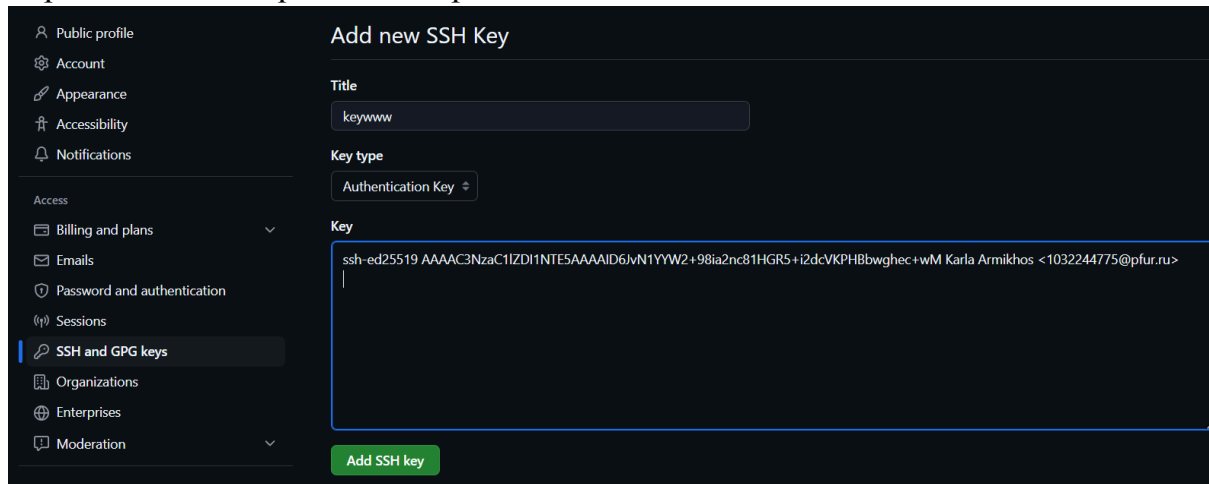


рис. 4.2.2

4.3 Сознание рабочего пространства и репозитория курса на основе шаблона

crear una carpeta


```
gkarmikhos@fedora:~$  
gkarmikhos@fedora:~$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютера"
```

рис. 4.3.1

Creamos un nuevo repositorio con el this template, modelo dado, siguiendo el link dado??

Required fields are marked with an asterisk (*).


Repository template

 **yamadharm/course-directory-student-template** ▾

Start your repository with a template repository's contents.

☐ **Include all branches**
Copy all branches from yamadharm/course-directory-student-template and not just the default branch.


Owner * **Repository name ***


 **ArmijosKarla** ▾ / **study_2024-2025_arh-pc**


✔ study_2024-2025_arh-pc is available.

Great repository names are short and memorable. Need inspiration? How about **super-octo-enigma** ?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

 You are creating a public repository in your personal account.

Create repository

рис. 4.3.2

клонировать папку с помощью ключа, созданного нашим репозиторием

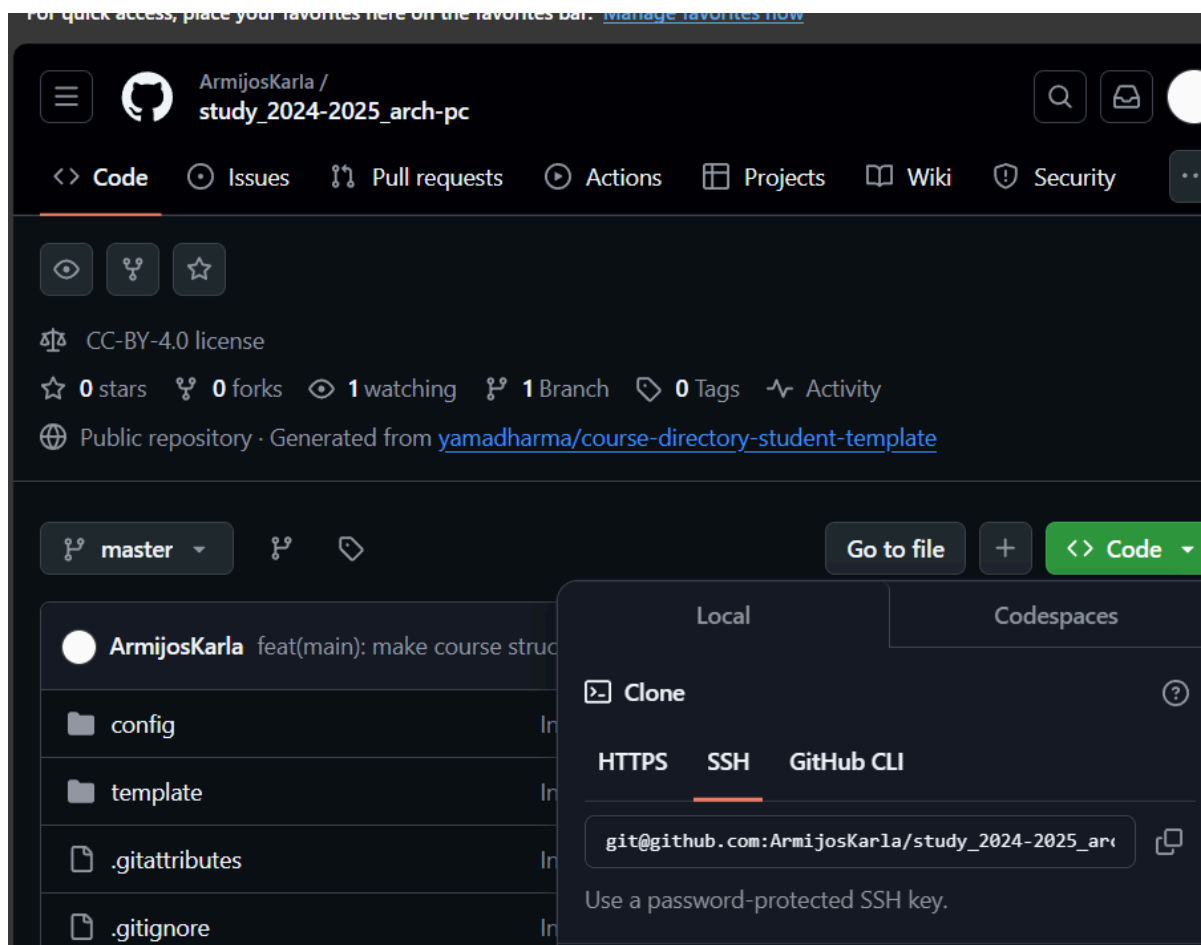


рис. 4.3.3

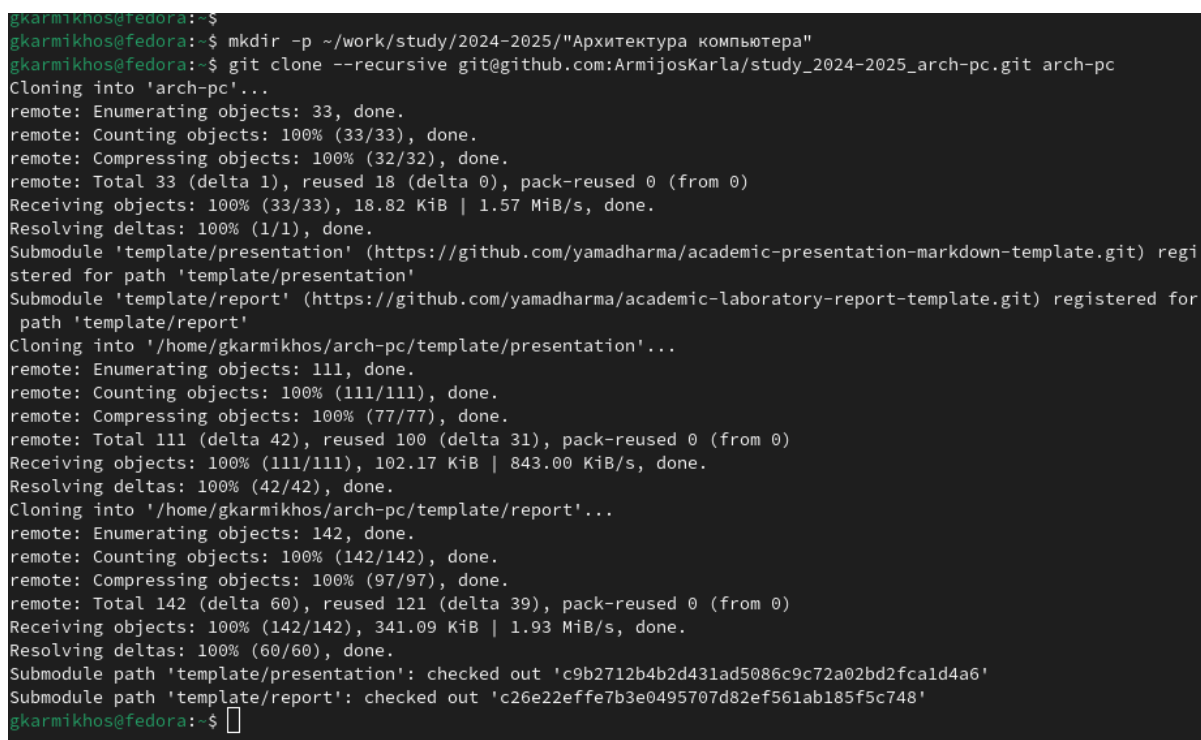


рис. 4.3.4

4.4 Настройка каталог курса

Перейдите в каталог курса и Удалите лишние файлы

```
gkarmikhos@fedora:~$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
```

рис. 4.4.1

Создайте необходимые каталоги command course

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
```

рис 4.4.2

команд make

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare       Generate directories structure
  submodule     Update submules
```

рис. 4.4.3

Отправьте файлы на сервер: помощью команды `git add . / f=git commit y git push`

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit
```

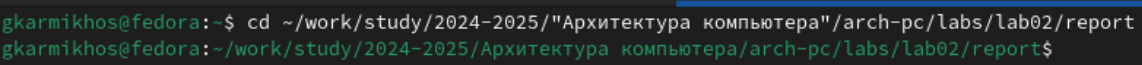
рис. 4.4.4

```
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master fdcdd82] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 293 bytes | 97.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:ArmijosKarla/study_2024-2025_arch-pc.git
 89adfc4..fdcd82 master -> master
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

рис 4.4.5

5 Задание для самостоятельной работы

создать una carpeta con el nombre lab 02

A terminal window with a dark background and green text. The first line shows the command 'cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab02/report' being entered. The second line shows the prompt 'gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report\$' after the command has been executed.

```
gkarmikhos@fedora:~$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab02/report
gkarmikhos@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$
```

рис. 5.1

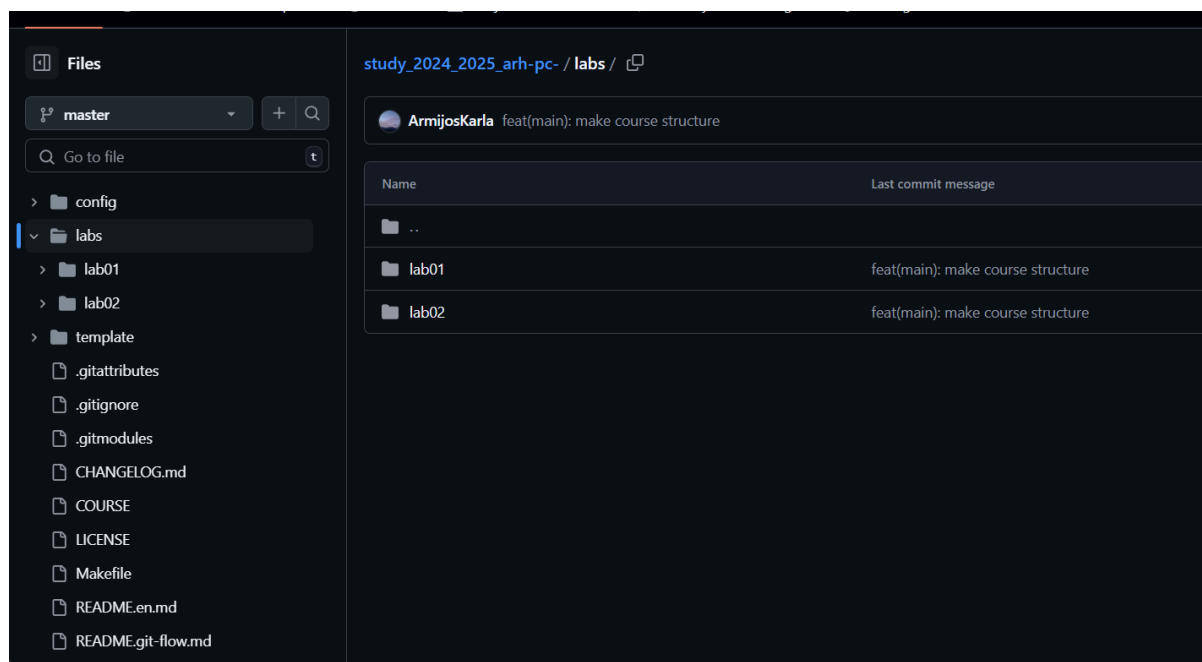
copiar los informes al espacio creado

рис. 5.2

subir a git

рис. 5.3

рис. 5.4



Войдите в Github и убедитесь, что документы находятся на месте.

Список используемой литературы

[Архитектура ЭВМ](#)

capturas de pantalla que teniai en fedora