

```

int a,b,c;
char vcheck[VERSIONSIZE];

gameaction = ga_nothing;

length = M_ReadFile(savename, &savebuffer);
save_p = savebuffer + SAVESTRINGSIZE;

// skip the description field
memset(&vcheck, 0, sizeof(vcheck));
sprintf(&vcheck, "version %d", VERSION);
if (strcmp(save_p, vcheck))
    return; // bad version
save_p += VERSIONSIZE;

gameskill = *save_p++;
gameepisode = *save_p++;
gamemap = *save_p++;
for (i=0 ; i<MAXPLAYERS ; i++)
    playeringamelist = *save_p++;

// load a base level
G_InitNew (gameskill, gameepisode, gamemap);

// get the times
a = *save_p++;
b = *save_p++;
c = *save_p++;
leveltime = (a<<16) + (b<<8) + c;

// dearchive all the modifications
P_UnArchivePlayers (i);
P_UnArchiveWorld (i);
P_UnArchiveThinkers (i);
P_UnArchiveSpecials (i);

if (*save_p != 0x1d)
    I_Error ("Bad savegame");

// done
Z_Free (savebuffer);

if (setsizeneeded)
    R_ExecuteSetViewSize (i);

// draw the pattern into the back screen
R_FillBackScreen (i);

```

```

void G_DoSaveGame (void)
{
    char name1[100];
    char name2[VERSIONSIZE];
    char* description;
    int length;
    int i;

    if (!M_CheckParm("-cdrom"))
        sprintf(name, "c:\\doomdata\\SAVEGAMENAME%ld.dsg", savegameslot);
    else
        sprintf (name, SAVEGAMENAME"%ld.dsg", savegameslot);
    description = savedescription;

    save_p = savebuffer = screens[1]+0x4000;

    memcpy (&save_p, description, SAVESTRINGSIZE);
    save_p += SAVESTRINGSIZE;
    memset (&name2, 0, sizeof(name2));
    sprintf (&name2, "version %d", VERSION);
    memcpy (&save_p, &name2, sizeof(name2));
    save_p += sizeof(name2);

    *save_p++ = gameskill;
    *save_p++ = gameepisode;
    *save_p++ = gamemap;
    for (i=0 ; i<MAXPLAYERS ; i++)
        *save_p++ = playeringamelist[i];
    *save_p++ = leveltime>>16;
    *save_p++ = leveltime>>8;
    *save_p++ = leveltime;

    P_ArchivePlayers (i);
    P_ArchiveWorld (i);
    P_ArchiveThinkers (i);
    P_ArchiveSpecials (i);
}

```

**Luis Pablo Carlos Tapia**  
**Carné: 202203257**

```

int episode, map )
{
    int i;

    if (paused)
    {
        paused = false;
        S_ResumeSound (i);
    }

    if (skill > sk_nightmare)
        skill = sk_nightmare;

    // This was quite messy with SPECIAL
    // Supposedly hacks to make the level
    // It might not work properly.
    if (episode < 1)
        episode = 1;

    if ( gamemode == retail )
    {
        if (episode > 4)
            episode = 4;
    }
    else if ( gamemode == shareware )
    {
        if (episode > 1)
            episode = 1; // only start episode 1 on shareware
    }
    else
    {
        if (episode > 3)
            episode = 3;
    }

    if (map < 1)
        map = 1;

    if ( (map > 9)
        && ( gamemode != commercial ) )
        map = 9;

    M_ClearRandom (i);

    if (skill == sk_nightmare || respawnparm)
        respawnmonsters = true;
    else
        respawnmonsters = false;

    if (fastparm || (skill == sk_nightmare && gameskill != sk_nightmare))
    {
        for (i=S_SARG_RUN1 ; i<=S_SARG_PAIN2 ; i++)
            states[i].tics >>= 1;
        mobinfo[MT_BRUISERSHOT].speed = 20*FRACUNIT;
        mobinfo[MT_HEADSHOT].speed = 20*FRACUNIT;
        mobinfo[MT_TROOPSHOT].speed = 20*FRACUNIT;
    }
    else if (skill != sk_nightmare && gameskill == sk_nightmare)
    {
        for (i=S_SARG_RUN1 ; i<=S_SARG_PAIN2 ; i++)
            states[i].tics <<= 1;
        mobinfo[MT_BRUISERSHOT].speed = 15*FRACUNIT;
        mobinfo[MT_HEADSHOT].speed = 10*FRACUNIT;
        mobinfo[MT_TROOPSHOT].speed = 10*FRACUNIT;
    }

    // force players to be initialized upon first level load
    for (i=0 ; i<MAXPLAYERS ; i++)
        players[i].playerstate = PST_REBORN;

    usergame = true; // will be set false if a demo
    paused = false;
    demoplayback = false;
    automapactive = false;
    viewactive = true;
    gameepisode = episode;
    gamemap = map;
    gameskill = skill;
}

```

# Tarea 2

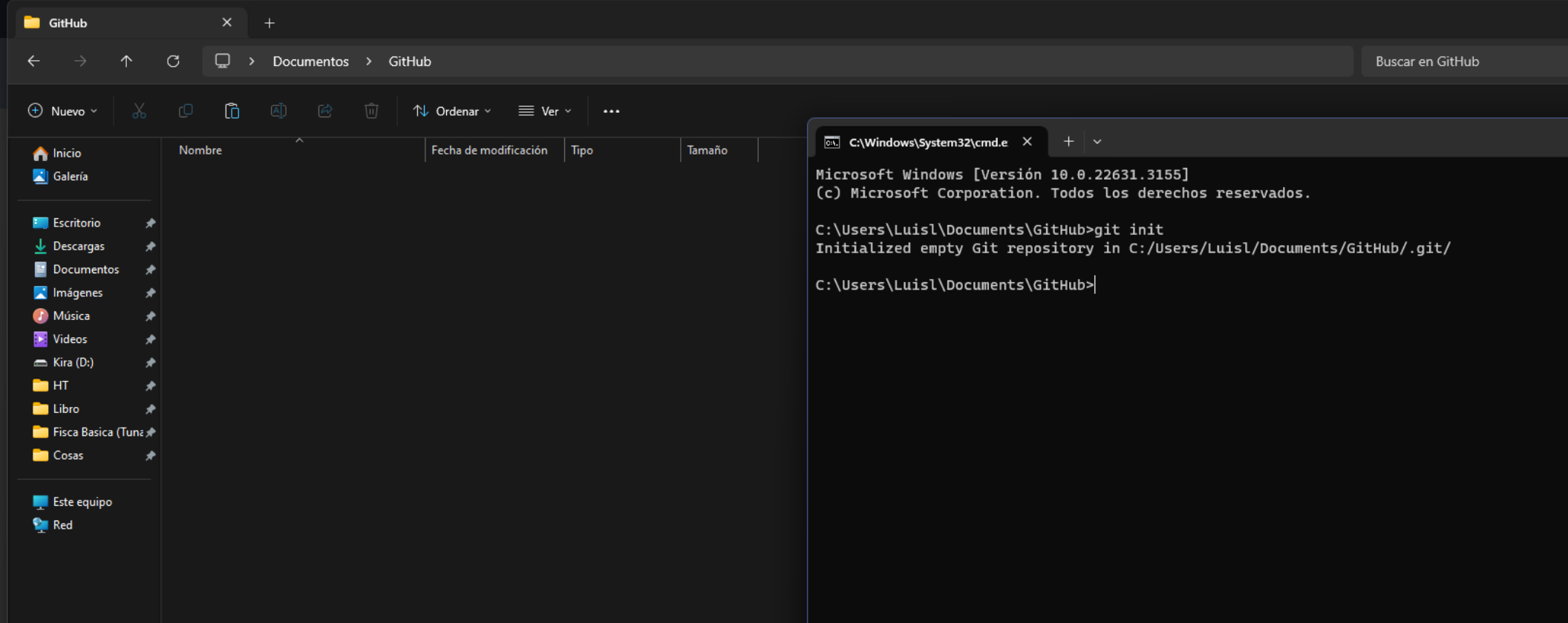
## LAB IPC1 Sección: G

```

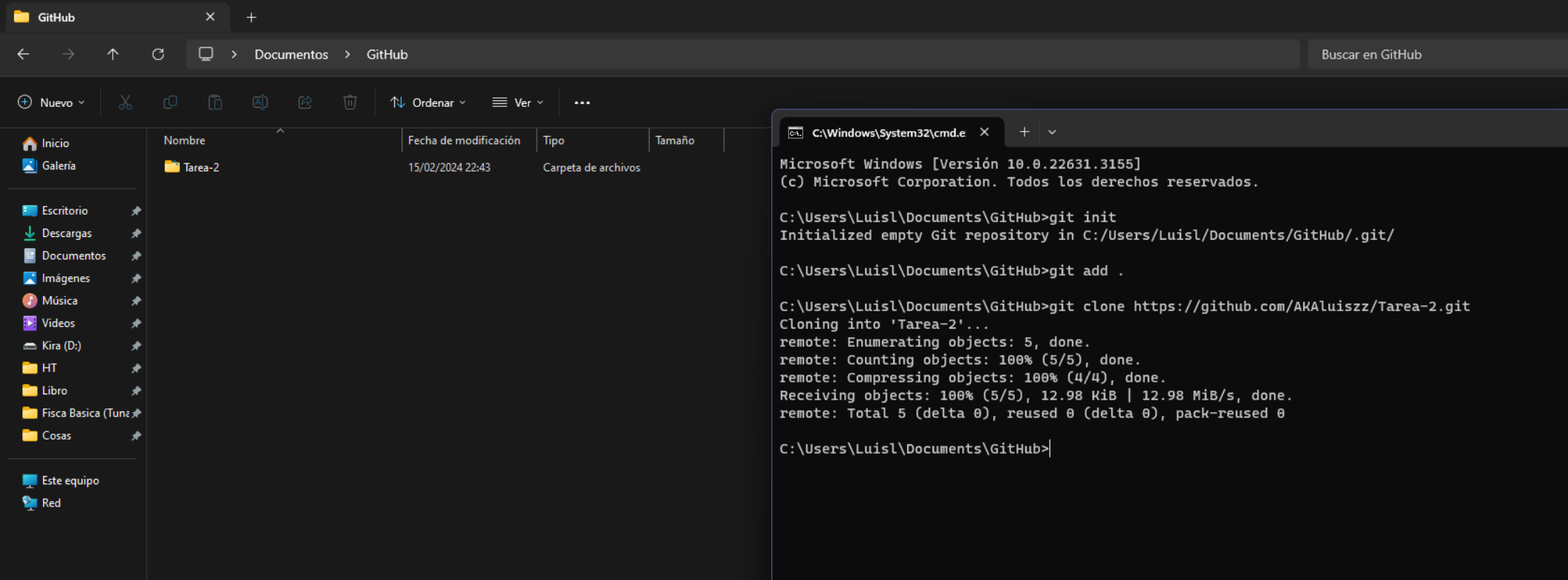
(SPR_PUNG,2,4,(NULL),S_PUNCH,0,0);
(SPR_PUNG,1,5,(A_ReFire),S_PUNCH,0,0);
(SPR_PISG,0,1,(A_WeaponReady),S_PISG,0,0);
(SPR_PISG,0,1,(A_Lower),S_PISTOLDOWN,0,0);
(SPR_PISG,0,1,(A_Raise),S_PISTOLDOWN,0,0);
(SPR_PISG,0,4,(NULL),S_PISTOL2,0,0);
(SPR_PISG,1,6,(A_FirePistol),S_PISTOL2,0,0);
(SPR_PISG,2,4,(NULL),S_PISTOL4,0,0);
(SPR_PISG,3,4,(A_ReFire),S_PISTOL,0,0);
(SPR_PLIGHT,1,1,(A_Light1),S_LIGHT1,0,0);
(SPR_PLIGHT,2,1,(A_WeaponReady),S_SGUN,0,0);
(SPR_PLIGHT,3,1,(A_Lower),S_SGUN,0,0);
(SPR_PLIGHT,4,1,(A_Raise),S_SGUNUP,0,0);
(SPR_PLIGHT,5,1,(A_Raise),S_SGUNUP,0,0);
(SPR_PLIGHT,6,1,(NULL),S_SGUN2,0,0);
(SPR_PLIGHT,7,1,(A_FireShotgun),S_SGUN,0,0);
(SPR_PLIGHT,8,1,(NULL),S_SGUN4,0,0);
(SPR_PLIGHT,9,1,(NULL),S_SGUN5,0,0);
(SPR_PLIGHT,10,1,(NULL),S_SGUN6,0,0);
(SPR_PLIGHT,11,1,(NULL),S_SGUN7,0,0);
(SPR_PLIGHT,12,1,(NULL),S_SGUN8,0,0);
(SPR_PLIGHT,13,1,(NULL),S_SGUN9,0,0);
(SPR_SHIT,0,7,(A_ReFire),S_SGUN,0,0);
(SPR_SHIT,32768,4,(A_Light1),S_SGUN,0,0);
(SPR_SHIT,32769,3,(A_Light2),S_LIGHT2,0,0);
(SPR_SHIT,0,1,(A_WeaponReady),S_DS,0,0);
(SPR_SHIT,0,1,(A_Lower),S_DS,0,0);
(SPR_SHIT,0,1,(A_Raise),S_DS,0,0);
(SPR_SHIT,0,3,(NULL),S_DS,0,0);
(SPR_SHIT,0,7,(A_FireShotgun2),S_DS,0,0);
(SPR_SHIT,1,7,(NULL),S_DS,0,0);
(SPR_SHIT,2,7,(A_CheckReload),S_DS,0,0);
(SPR_SHIT,3,7,(A_OpenShotgun2),S_DS,0,0);
(SPR_SHIT,4,7,(NULL),S_DS,0,0);
(SPR_SHIT,5,7,(A_LoadShotgun2),S_DS,0,0);
(SPR_SHIT,6,6,(NULL),S_DS,0,0);
(SPR_SHIT,7,6,(A_CloseShotgun2),S_DS,0,0);
(SPR_SHIT,0,5,(A_ReFire),S_DS,0,0);
(SPR_SHIT,1,7,(NULL),S_DS,0,0);
(SPR_SHIT,0,3,(NULL),S_DS,0,0);
(SPR_SHIT,32776,5,(A_Light1),S_DS,0,0);
(SPR_SHIT,32777,4,(A_Light2),S_LIGHT2,0,0);
(SPR_CHGG,0,1,(A_WeaponReady),S_CH,0,0);
(SPR_CHGG,0,1,(A_Lower),S_CH,0,0);
(SPR_CHGG,0,1,(A_Raise),S_CH,0,0);
(SPR_CHGG,0,4,(A_FireCGun),S_CH,0,0);
(SPR_CHGG,1,4,(A_FireCGun),S_CH,0,0);
(SPR_CHGG,1,0,(A_ReFire),S_CH,0,0);
(SPR_CHGF,32768,5,(A_Light1),S_LIGHT1,0,0);
(SPR_CHGF,32769,5,(A_Light2),S_LIGHT2,0,0);
(SPR_MISG,0,1,(A_WeaponReady),S_MISS,0,0);
(SPR_MISG,0,1,(A_Lower),S_MISS,0,0);
(SPR_MISG,0,1,(A_Raise),S_MISS,0,0);
(SPR_MISG,1,8,(A_GunFlash),S_MISS,0,0);
(SPR_MISG,1,12,(A_FireMissile),S_MISS,0,0);
(SPR_MISG,1,0,(A_ReFire),S_MISS,0,0);
(SPR_MISF,32768,3,(A_Light1),S_MISS,0,0);
(SPR_MISF,32769,4,(NULL),S_MISS,0,0);
(SPR_MISF,32770,4,(A_Light2),S_MISS,0,0);
(SPR_MISF,32771,4,(A_Light2),S_MISS,0,0);
(SPR_SAWG,2,4,(A_WeaponReady),S_SAW,0,0);
(SPR_SAWG,3,4,(A_WeaponReady),S_SAW,0,0);
(SPR_SAWG,2,1,(A_Lower),S_SAW,0,0);
(SPR_SAWG,0,4,(A_Saw),S_SAW2,0,0);
(SPR_SAWG,1,4,(A_Saw),S_SAW3,0,0);
(SPR_SAWG,1,0,(A_ReFire),S_SAW,0,0);
(SPR_PLSG,0,1,(A_WeaponReady),S_PL,0,0);
(SPR_PLSG,0,1,(A_Lower),S_PLASMA,0,0);
(SPR_PLSG,0,1,(A_Raise),S_PLASMA,0,0);
(SPR_PLSG,0,3,(A_FirePlasma),S_PLASMA,0,0);
(SPR_PLSG,1,20,(A_ReFire),S_PLASMA,0,0);
(SPR_PLSF,32768,4,(A_Light1),S_LIGHT1,0,0);
(SPR_PLSF,32769,4,(A_Light1),S_LIGHT1,0,0);
(SPR_BFGG,0,1,(A_WeaponReady),S_BFG,0,0);
(SPR_BFGG,0,1,(A_Lower),S_BFG,0,0);
(SPR_BFGG,0,1,(A_Raise),S_BFG,0,0);
(SPR_BFGG,0,20,(A_BFGSound),S_BFG,0,0);
(SPR_BFGG,1,10,(A_GunFlash),S_BFG,0,0);
(SPR_BFGG,1,10,(A_FireBFG),S_BFG,0,0);
(SPR_BFGG,1,20,(A_ReFire),S_BFG,0,0);
(SPR_BFGF,32768,11,(A_Light1),S_BFG,0,0);
(SPR_BFGF,32769,6,(A_Light2),S_LIGHT2,0,0);
(SPR_BLWD,2,8,(NULL),S_BLOOD2,0,0);
(SPR_BLWD,1,8,(NULL),S_BLOOD3,0,0);
(SPR_BLWD,0,8,(NULL),S_BLOOD4,0,0);

```

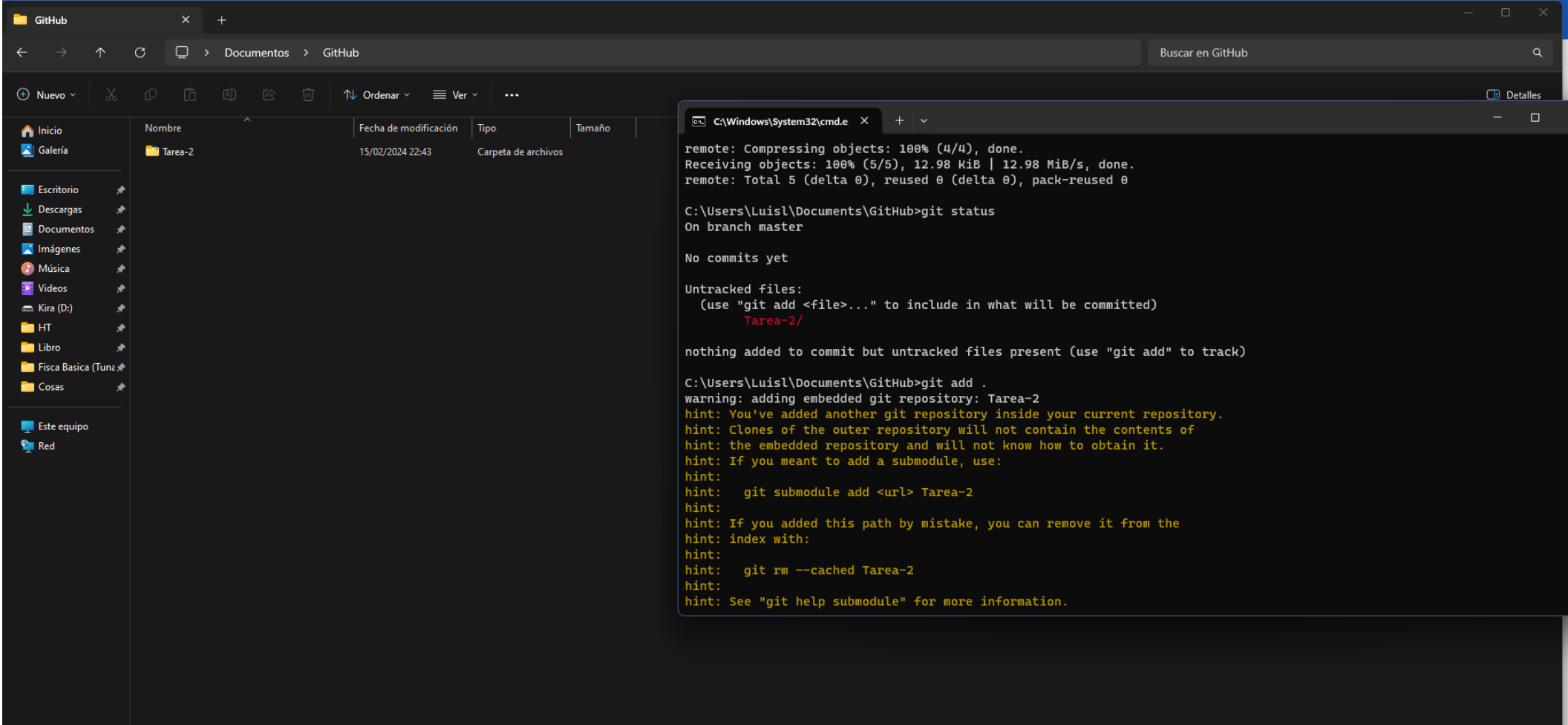




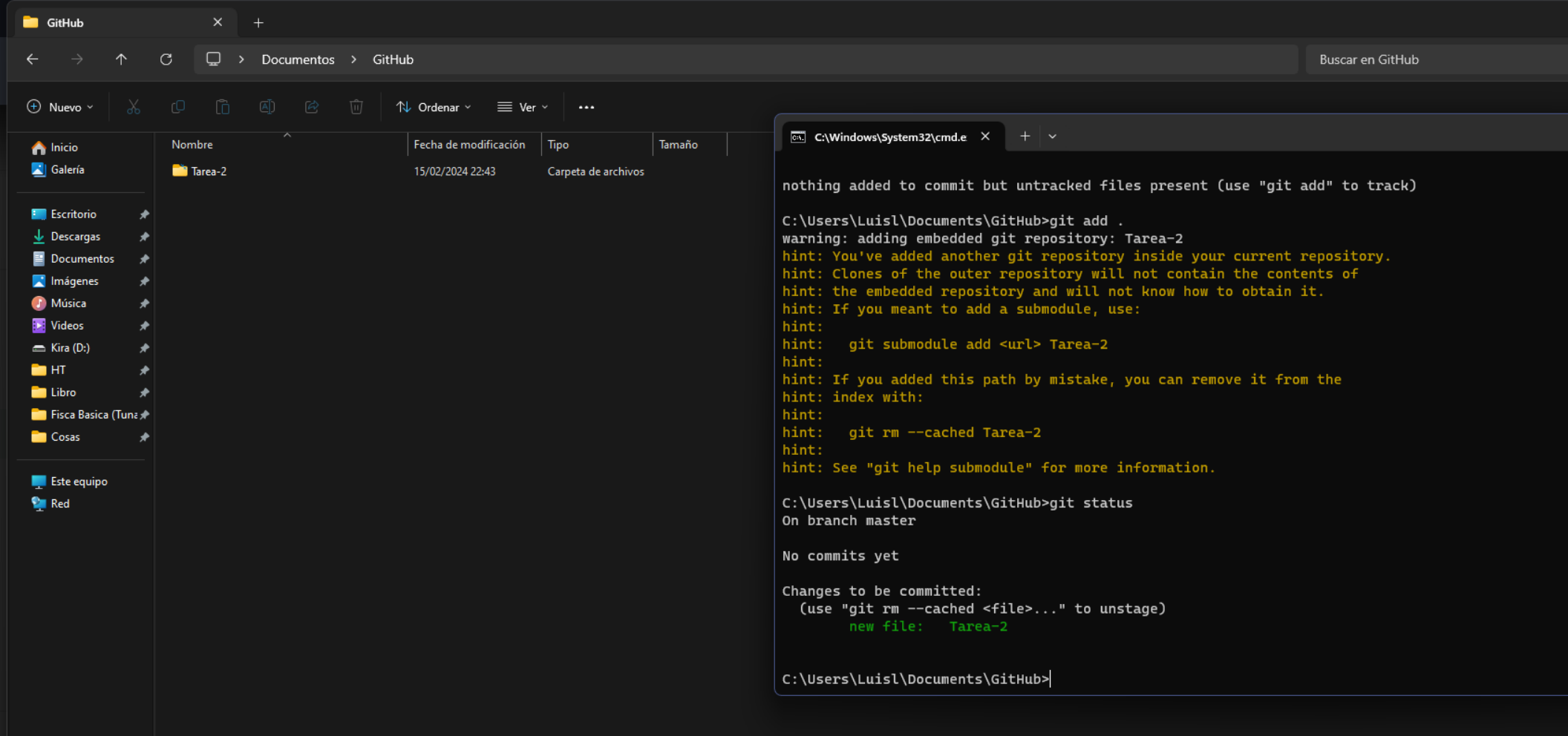
Comando 1: Git Init.



## Comando 2: Git Clone.

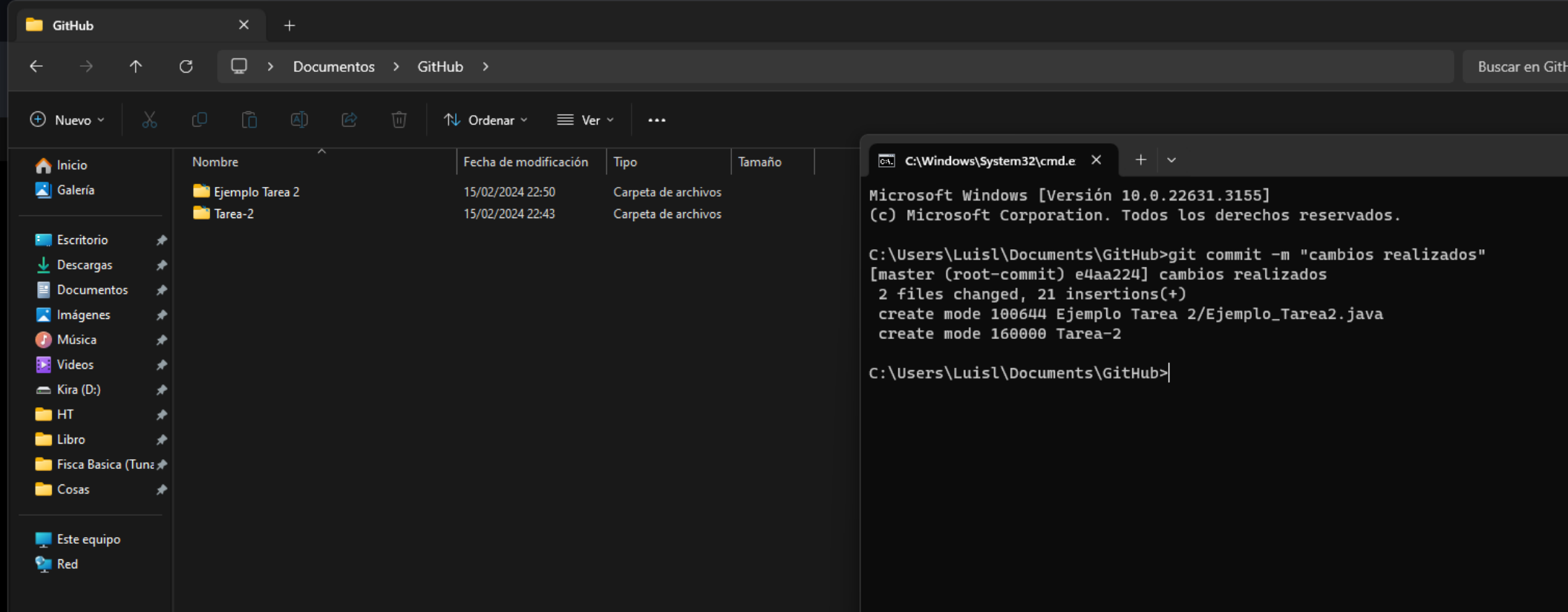


Comando 3: Git Add .



## Comando 4: Git Status.





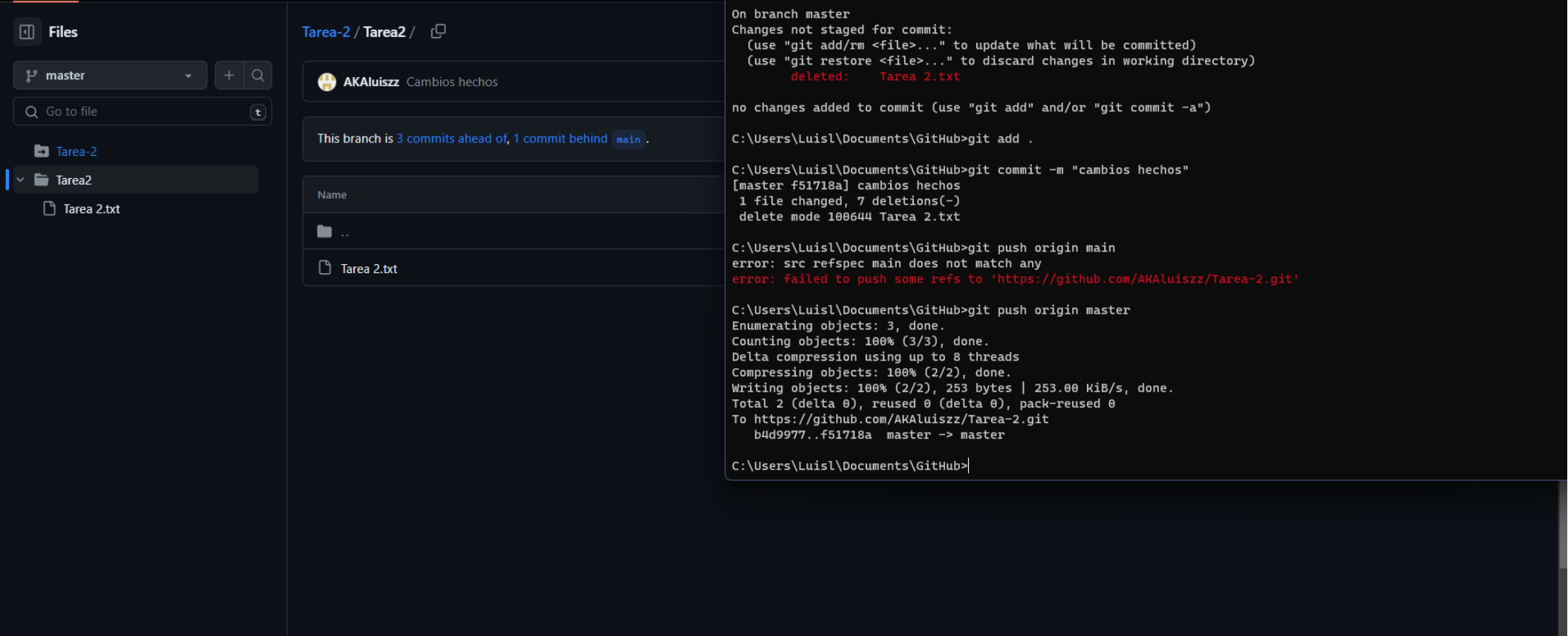
Comando 5: Git Commit.

```
C:\Users\Luisl\Documents\GitHub>git branch
* master

C:\Users\Luisl\Documents\GitHub>|
```

Comando 5: Git branch.





Comando 5: Git push.

Link del repo: <https://github.com/AKAluiszz/Tarea-2.git>