Nombre: Diego Sandoval Castellanos

Matrícula: 460293411

Materia: MOD y MET P EL D de software

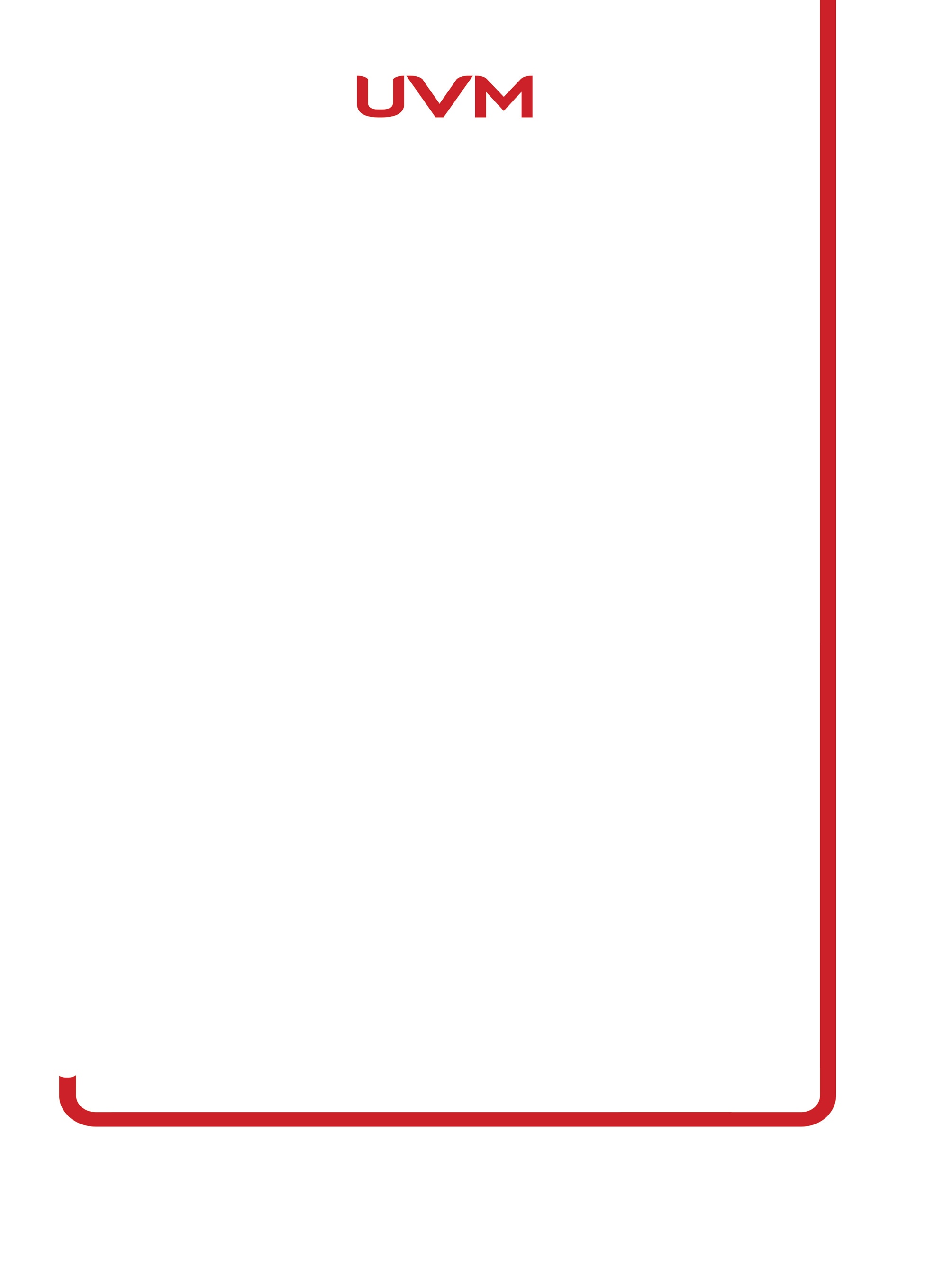
Profesor: Julio Cesar Villa López

Campus: Zapopan

Fecha de entrega: 13/06/2024

Uso y configuración de GIT.

[Subtítulo del documento]



Investiga y contesta:

1. ¿Cómo se establece el nombre de usuario en git? Para establecer el nombre de usuario globalmente en Git, puedes usar el comando git config --global user.name "Tu Nombre". Si deseas establecerlo solo para un repositorio específico, omite la opción --global: git config user.name "Tu Nombre".
2. ¿Cómo se establece el correo del usuario en git? Para establecer el correo del usuario globalmente en Git, puedes usar el comando git config --global user.email "tu.email@example.com". Para establecerlo solo en un repositorio específico, omite la opción --global: git config user.email "tu.email@example.com".
3. ¿Cómo se activa el coloreado de salida en git? Para activar el coloreado de salida en Git, puedes usar el comando git config --global color.ui auto.
4. ¿Cómo se muestra el estado original en los conflictos de git? Para mostrar el estado original en los conflictos de Git, puedes usar el comando git checkout --theirs <ruta\_del\_archivo> o git checkout --ours <ruta\_del\_archivo>, dependiendo de si deseas mantener los cambios de la rama que estás fusionando (--theirs) o los de tu rama actual (--ours).
5. ¿Cómo se muestra la configuración? Para mostrar la configuración de Git, usa el comando git config --list.
6. ¿Cómo se crea un repositorio en GIT? Para crear un repositorio en Git, navega al directorio donde quieres crear el repositorio y usa el comando git init.
7. ¿Cómo se muestra el estado de un repositorio en git? Para mostrar el estado de un repositorio en Git, usa el comando git status.
8. ¿Cómo se muestra las diferencias entre versiones en git? Para mostrar las diferencias entre versiones en Git, usa el comando git diff. Para ver las diferencias entre dos commits específicos: git diff <commit1> <commit2>.
9. ¿Cómo se puede volver a un directorio de trabajo o versión anterior? Para volver a un commit anterior, puedes usar el comando git checkout <commit>. Para resetear el estado del repositorio a un commit anterior (afectando el historial), usa git reset --hard <commit>.
10. ¿A qué se refiere con un pull en git? Un git pull es el comando que se usa para actualizar el repositorio local con los cambios del repositorio remoto. Combina dos comandos: git fetch (para descargar los cambios) y git merge (para fusionar los cambios): git pull.
11. ¿A qué se refiere con un push en git? Un git push es el comando que se usa para enviar tus cambios locales al repositorio remoto: git push.

Practica:

* Crea un repositorio local en Git
* Comparte ese repositorio con cada uno de tus compañeros de equipo del directorio remoto.
* Que cada compañero tenga su escritorio remoto y añada los cambios necesarios.
* Añadan los cambios al stage.
* Registren los cambios.
* Suban los cambios al remoto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* Realiza las siguientes operaciones sobre la rama:
  + Crea su rama.
  + Ver en que rama se encuentra.
  + Cambiar de rama.
  + Ver los cambios entre ramas.
  + Fusionar ramas
  + Eliminar rama.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Cada compañero deberá de mostrar su rama y su usuario además de que comparte el mismo repositorio.
* Para un pull request:
  + Crear la rama.
  + Añada su nombre de usuario en README.md en el apartado Colaboradores.
  + Sube los cambios en local.
  + Subir los cambios.
  + Ir a github y solicitar una pull request.
  + Agregar un comentario.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Codigo.

Ser requiere que se genere lo siguiente:

Codificación de botón de eliminar y botón limpiar.

Botón eliminar:

Debe de buscar en base al código y si no existe mencionar que asi es.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Botón Limpiar:

Debe de limpiar todos los campos de texto, esto lo realiza llamando una función que se debe de generar en el código, es decir no escribimos el código en el botón si no que generamos una función y se manda llamar.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Link:

<https://github.com/DiegoThere/Proyecto_tienda.git>

BD:

Deberás mostrar el antes y el después.

Interfaz de usuario gráfica, Texto, Aplicación, Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Abre tu plantilla y genera las actividades que consideres necesarias para los pasos anteriores.
* Genera las historias de usuario.

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 5** | **Usuario:** Empleado | |
| **Nombre historia: Eliminar de producto** | | |
| **Prioridad en negocio:**  Alta | | **Riesgo en desarrollo:**  Media |
| **Puntos estimados: 5** | | **Iteración asignada: 1** |
| **Programador responsable:** Diego | | |
| **Descripción:**  El sistema debe poderse conectar a la base de datos con la tabla productos, el sistema debe permitir eliminar productos en la tabla productos, se debe validar el funcionamiento de la interfaz | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 6** | **Usuario:** Empleado | |
| **Nombre historia: Elimpiar de producto y repositorio** | | |
| **Prioridad en negocio:**  Alta | | **Riesgo en desarrollo:**  Media |
| **Puntos estimados:** 4 | | **Iteración asignada: 1** |
| **Programador responsable:** Diego | | |
| **Descripción:**  El sistema debe poder limpiar la pantalla donde se están registrando los datos, verificar funcionamiento de la interfaz, crear repositorio github | | |
| **Observaciones:** | | |

* Pasamos a codificar.

A entregar:

* Planificación e historias de usuario en repositorio que compartirás con el docente.
* Captura de pantalla de cada uno de los pasos solicitados.
* BD
* Interfaz funcionando.
* Liga o enlace a tu repositorio.