

Objetivo

OET7. Utilizar un sistema de control de versiones para la gestión de los archivos producidos durante el proceso de desarrollo de software.

Enunciado

Usted ha sido contratado por una importante empresa de desarrollo de programas para ser parte de un equipo de desarrollo que debe entregar una funcionalidad para un producto que ya fue iniciado por otros desarrolladores, pero que no se encuentra completo. Debe emplear el sistema de control de versiones Git para copiar y descargar apropiadamente el proyecto y agregar e integrar las funcionalidades desarrolladas por usted y su equipo de trabajo.

La alcaldía de Santiago de Cali desea implementar un sistema de pago de parqueadero público que permita cobrar el parqueo de vehículos en zonas públicas. Los dispositivos donde las personas podrán pagar se dispondrán en la calle y cada persona tendrá únicamente que indicar qué tipo de vehículo ha parqueado (Moto o Carro), el dispositivo le emitirá una tarjeta con la cual el conductor pagará por el tiempo utilizado al regresar por su vehículo.

El programa actualmente luce como se muestra en la Figura 1.



Figura 1. Programa inicial cuando empieza su ejecución.

Hasta el momento, cuando el usuario digita el tipo de vehículo, el prototipo asigna las 7:00 como hora inicial. Esta se visualiza en el primer campo de izquierda a derecha, mientras los demás permanecen en blanco. Posteriormente, cuando el conductor regrese por su vehículo y presione el botón Calcular, se le asigna automáticamente las 12:00 como hora final. Inmediatamente el programa calcula el costo de parqueo escribiendo el resultado en el campo Valor a Pagar.



Figura 2. Programa después de generar el valor a pagar.

Se le pide que pruebe el programa en su estado actual y que verifique todas las funcionalidades anteriores.

Asimismo, la empresa cree que el prototipo actual no es el adecuado, por tal le solicita realizar ciertas mejoras:

1. Al usuario elegir el tipo de vehículo, el prototipo debe generar aleatoriamente una hora de la mañana entre 7:00 y 11:59. Igualmente, al presionar el botón Calcular, se ha de generar aleatoriamente una hora entre 12:00 y 21:59 que se mostrará en el campo Hora Final.
2. La presentación de la herramienta ha de mejorar, por tal cada campo debe tener su título. Tipo de Vehículo, Hora de Inicio y Hora Final.
3. Para conocer de qué trata la herramienta se debe incluir un título en la ventana principal que diga Parqueo Público.
4. Ya que el valor a pagar es un monto de dinero, su formato debe ir acorde con el formato de moneda.

Ustedes deben formar 6 equipos de 4 personas cada uno. Cada equipo trabajará en las cuatro mejoras indicadas anteriormente. Se elegirá a un líder que deberá velar por que todo funcione correctamente mientras que los demás integrantes se deben poner de acuerdo para que cada uno trabaje en un cambio diferente.

Instrucciones

1. Ingresar a <https://github.com/> y crear una cuenta.

- 2.
3. Formar los equipos de 4 personas cada uno, ponerle un nombre al equipo y designar un líder. En la siguiente tabla aparecerá el nombre de cada equipo en las celdas gris oscuro y debajo los integrantes, dejando el líder en la celda gris claro.

SAVE THE WORLD BY OVERLOADING IT WITH FUN CARLOS JIMMY BRIGADE	[:)]	Equipo Estrellita	Los Chigüiros	Otro grupo (G6)	Equipo 5
<u>Juan José Bohórquez Jaramillo</u>	<u>Samuel Viviescas</u>	Carolina Pasuy	Kevin Mera	Jose Alejandro García	Juan Pablo Sanin
<u>Juan Fernando Martinez (Lider)</u>	<u>Paola Osorio</u>	<u>Juan Pablo Ramos</u>	Juan Esteban Caicedo	Anderson Rueda (Líder)	El poderoso GioGio
<u>Carlos Jimmy Pantoja</u>	<u>Sean</u>	<u>Anderson cardenas</u>	Admin: Julian Riascos		Juan Camilo Zorrillla
<u>Bryan Guapacha</u>	<u>Diego Hernandez</u>	Juan Palta	John Kennedy		juanserodr iguez16

4. Las nuevas funcionalidades, con los cambios necesarios para llevarlas a cabo, se detallan a continuación. Escriba en la columna responsables los nombres de las personas que estarán encargadas de desarrollar esa modificación específica.

#F	Clases y Métodos a cambiar	Modificación esperada	Responsables
1.	Clase: Parqueadero Métodos: marcarHoraInicio marcarHoraFinal	Realizar cambios en la asignación para las dos variables horaInicio y minutoInicio. horaInicio = (int)(7+Math.random()*5); minutoInicio = (int)(Math.random()*60); horaInicio = (int)(12+Math.random()*10); minutoInicio = (int)(Math.random()*60);	G1: Juan José Bohórquez Grupo 2: Samuel Viviescas Equipo 5: Juan Camilo Zorrilla G4 John KennedyJ Grupo Estrellita: JuanPalta G6: Anderson Rueda
2..	Clase: ParqueoPublico Método: start	Adicionar la instrucción stage.setTitle("Parqueo Público");	Grupo Estrellita: Caro Grupo 4: Kevin Mera Grupo 2: Paola Osorio

			G6: Jose Alejandro García G5: Juan Sebastian Rodriguez Salazar
3.	Clase: ParqueoPublicoController Método: calcularButtonPresionado	Realizar cambios en el formato del campo de texto valorAPagarTextField.setText(CURRENCY.format(valorAPagar));	Grupo 4: Julián Riascos Grupo estrellita: Juan Pablo Ramos Grupo 2: Diego Hernandez G5: Juan Pablo Sanin G6: Anderson Rueda
4.	Archivo: ParqueoPublico.fxml	Incluir las siguientes líneas <Label text="Vehículo" GridPane.halignment="CENTER" /> <Label text="Hora de Inicio" GridPane.columnIndex="1" GridPane.halignment="CENTER" /> <Label text="Hora Final" GridPane.columnIndex="2" GridPane.halignment="CENTER" />	G2: Sean Grupo estrellita: Anderson G6: Jose Alejandro García G4: Juan Esteban Caicedo G5: Giovanni Mosquera G1: Bryan Guapacha

- El líder del proyecto debe hacer un fork del repositorio <https://github.com/andresaristi/PublicParking>. Esto lo consigue entrando al enlace anterior y accediendo a la opción Fork de la esquina superior derecha. Quien hace el fork debe estar autenticado en github y de esta manera se creará su propia copia del proyecto como repositorio en su propia cuenta.
- Invitar a colaborar al repositorio del líder a todos los demás integrantes de su equipo de trabajo. Para esto debe dirigirse a la pestaña Settings del menú superior del repositorio y luego entrar a la opción Collaborators del menú izquierdo. Luego de ser invitados, cada uno de los colaboradores recibirá un correo con la invitación a colaborar. Esta invitación debe ser aceptada y entonces todos los integrantes del equipo tendrán acceso al mismo repositorio.
- Puede abrir la consola de Git llamada Git Shell, aunque se recomienda usar Windows PowerShell. Pruebe los comandos **pwd** (que indica en qué directorio se encuentra la consola), **cd directorio** (para entrar a un subdirectorio llamado *directorio* del directorio donde se encuentra), **cd ..** (para salirse del directorio actual y quedarse en el directorio padre). Luego de probar los comandos anteriores, en la consola debe ubicarse en el directorio donde trabajará su proyecto, podría ser directamente en el directorio **d**, o en **c/User/cédula/Documents** o el directorio que usted elija.
- Clonar el proyecto del líder que se encuentra en el repositorio en GitHub. El repositorio del líder debe tener una url similar a ésta: <https://github.com/nombreusuariodellider/PublicParking>. Debe ingresar en la consola el comando que le permite clonar dicho repositorio: `git clone https://github.com/nombreusuariodellider/PublicParking`. Con esto se descarga todo el proyecto del repositorio al directorio actual.
- Entrar al directorio del proyecto recién clonado: `cd PublicParking`. Luego escriba el comando: `ls -a`, para ver un listado de todos los archivos descargados y luego: `git status` para conocer el estado actual del repositorio y sus modificaciones.
- Abrir eclipse y luego importar este proyecto a través de la opción Import... -> Git -> Projects from Git. Luego elegimos la opción: Existing local repository, después clic en el botón Add... y finalmente en el botón Browse... para seleccionar el directorio del proyecto clonado. Al finalizar este paso, su proyecto ha quedado importado en eclipse, listo para ser ejecutado y modificado. Note que varios de los elementos del proyecto en la ventana Package Explorer tienen un ícono de cilindro naranja y que los archivos están anteceditos por el símbolo >. Lo anterior indica que su proyecto está enlazado con un repositorio de Git.
- Ejecutar el proyecto y verifique que las funcionalidades descritas al inicio de este documento funcionan perfectamente.
- Desarrollar la funcionalidad que le corresponde, de acuerdo con la tabla de división de responsabilidades.

13. Ejecute el proyecto y pruebe que sus cambios funcionan bien.
14. Luego de hacer los cambios asignados. Debe agregar los archivos modificados al stage. Se sugiere ejecutar el comando: `git status`, para confirmar qué archivos ha cambiado durante el desarrollo de los cambios, y luego agregarlos al stage a través del comando: `git add archivo modificado`.
15. Una vez agregados los archivos al stage, usted debe confirmar su envío al repositorio local utilizando el comando: `commit -m "write down here an appropriate description, in english, related to the change"`.
16. Luego usted debe sincronizar el proyecto remoto con su proyecto local, pero en su repositorio local haciendo pull. Pull es un comando que hace primero fetch y luego merge. Usted debe entonces ejecutar el comando: `git pull`, y resolver los conflictos si se presentan.
17. Una vez el pull ha sido ejecutado exitosamente, se debe proceder a enviar los cambios al repositorio remoto con el comando: `git push`.

Recuerde el orden clave:

(1) add (2) commit (3) pull (4) push