

**Proyecto para el desarrollo de la Pagina Web para la
Gestión de Acciones
Flujo de Trabajo**

Versión 1.0

Gestión de Acciones	Versión: 1.0
Documento de Flujo de Trabajo	Fecha: 20/11/2024

Historial de Revisiones

Fecha	Versión	Descripción	Autor
14/10/2024	1.0	Definición del Flujo de Trabajo	Grupo Al filo del Éxito: Defaz Joel Haro Rodrigo Mendoza Miguel Trocellier Michael

Gestión de Acciones	Versión: 1.0
Documento de Flujo de Trabajo	Fecha: 20/11/2024

Tabla de Contenidos

1	Equipo	4
1.1	Nombre del equipo	4
1.2	Slogan	4
1.3	Logo	4
1.4	Miembros del Equipo	4
2	Proyecto	5
2.1	Descripción	5
3	Flujo de Trabajo	5
3.1	Objetivo del Flujo de Trabajo	5
3.2	Flujo de Trabajo en Git Hub	5
3.3	Etapas del Flujo	6
3.3.1	Planeación	6
3.3.2	Diseño	6
3.3.3	Codificación	7
3.3.4	Pruebas	7
3.4	Roles y Responsabilidades	8
3.5	Herramientas y Recursos	8
3.6	Iteración o Ciclo	8
3.7	Dinámica del Equipo:	9

Gestión de Acciones	Versión: 1.0
Documento de Flujo de Trabajo	Fecha: 20/11/2024

1 Equipo

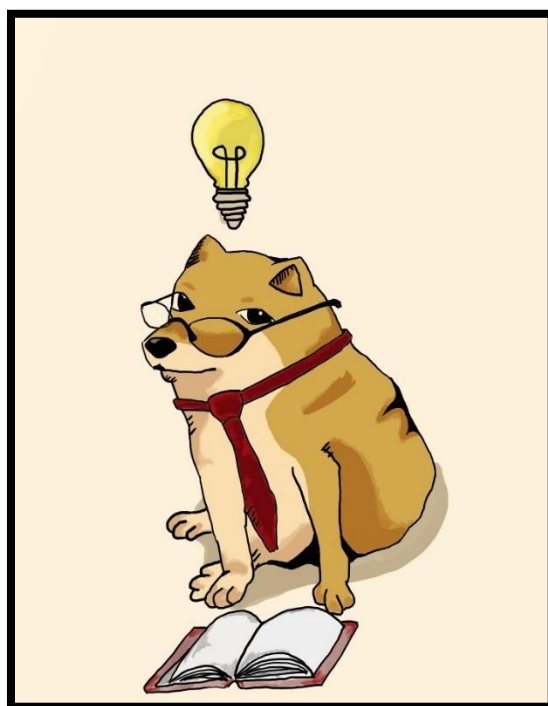
1.1 Nombre del equipo

Al Filo del Éxito

1.2 Slogan

Reprobado, pero no derrotado

1.3 Logo



1.4 Miembros del Equipo

Nombre	Habilidades/Descripción	Contacto
Defaz Joel		jose.defaz@epn.edu.ec
Haro Rodrigo		rodrigo.haro@epn.edu.ec
Mendoza Miguel		miguel.mendoza@epn.edu.ec
Trocellier Michael		

Gestión de Acciones	Versión: 1.0
Documento de Flujo de Trabajo	Fecha: 20/11/2024

2 Proyecto

2.1 Descripción

El presente proyecto tiene como objetivo desarrollar una herramienta eficiente para gestionar y analizar inversiones en acciones. Este sistema permitirá registrar la compra de acciones en una fecha determinada, especificando la cantidad adquirida y el valor correspondiente. A partir de dicha información, el sistema será capaz de calcular automáticamente el porcentaje de ganancia o pérdida en función de la fecha actual y las fluctuaciones del mercado. Asimismo, proporcionará el equivalente monetario de estas variaciones, facilitando a los usuarios una visión clara y precisa de sus rendimientos financieros.

3 Flujo de Trabajo

3.1 Objetivo del Flujo de Trabajo

Gestionar y desarrollar un sistema que permita registrar y analizar compras de acciones.

3.2 Flujo de Trabajo en Git Hub

Para el flujo de trabajo lo que se ha decidió con el equipo es realizar un flujo de trabajo tal y como se indica en la imagen 1, en la cual consta de las siguientes ramas:

- **Main/master:** La rama principal donde se colocará la versión estable de la aplicación después de realizar las pruebas respectivas, constará con el tag del código de la versión estable.
- **Docs:** La rama en la que se colocarán los archivos de configuración correspondientes, constará con el tag de documentación de la versión estable.
- **Develop:** rama en la cual se juntarán las diferentes características de la aplicación.
- **Test:** Ramas temporales para realizar las pruebas de las nuevas características añadidas o errores corregidos, se eliminarán después de realizar la prueba respectiva.
- **Module:** Ramas para cada módulo de la aplicación, en la que trabajará cada miembro del equipo con su módulo asignado.
- **Feature:** Ramas temporales para añadir nuevas funcionalidades o corregir errores en cada módulo, se eliminarán después de cumplir su objetivo.

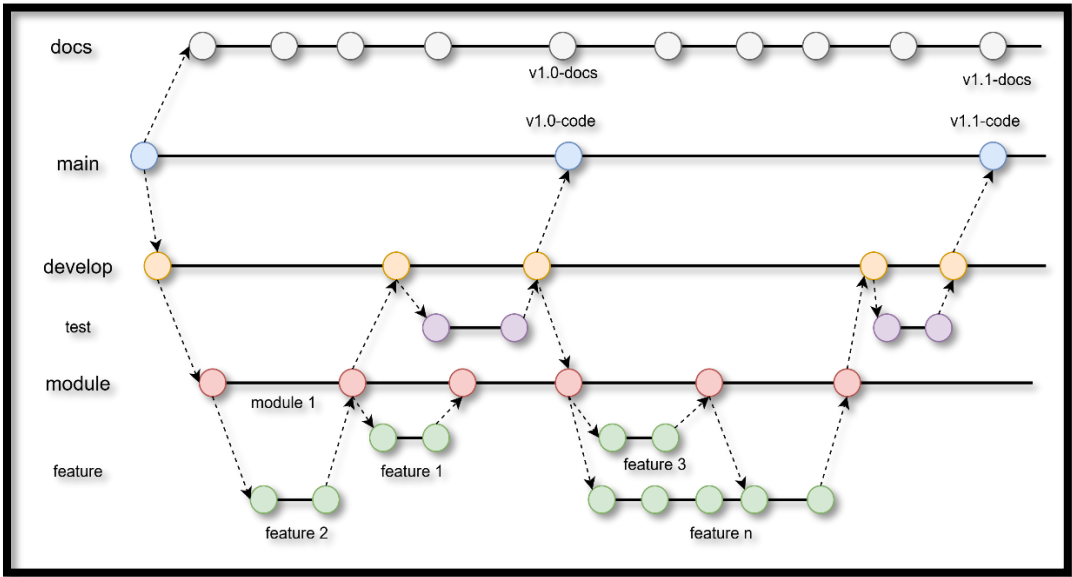


Imagen 1. Flujo de Trabajo

Gestión de Acciones	Versión: 1.0
Documento de Flujo de Trabajo	Fecha: 20/11/2024

3.3 Etapas del Flujo

3.3.1 *Planeación*

Se realizan actividades clave para asegurar que el proyecto esté alineado con las necesidades del cliente y que el equipo tenga claridad sobre las prioridades.

- **Definición de historias de usuario:** El equipo y el cliente redactan historias de usuario claras y comprensibles, capturando los requerimientos sin entrar en detalles técnicos.
- **Priorización de historias:** El cliente prioriza las historias según su valor de negocio, enfocando el desarrollo en las funcionalidades más críticas.
- **Estimación de tiempos:** Se estima el tiempo necesario para implementar cada historia, ayudando a identificar obstáculos y medir el esfuerzo requerido.
- **Entradas y salidas:**
 - **Entradas:** Requerimientos del cliente, historias de usuario iniciales, disponibilidad y capacidad del equipo, y recursos disponibles.
 - **Salidas:** Historias priorizadas, estimaciones de esfuerzo, plan de iteraciones, cronograma, definición de alcance y lista de riesgos.
- **Responsables:**
 - **Developer:** Estima el esfuerzo y evalúa la viabilidad técnica.
 - **Coach:** Facilita la comunicación y asegura que se sigan los principios de XP.
 - **Tracker:** Verifica que las estimaciones sean realistas y evita sobrecargar al equipo.
 - **Tester:** Define los criterios de aceptación y asegura la calidad de las funcionalidades implementadas.

3.3.2 *Diseño*

En la etapa de diseño, se busca crear una estructura simple y flexible para desarrollar el sistema de manera ágil y adaptable. En XP, el diseño no se enfoca en documentos extensos o detalles complejos, sino en una base clara que permita avances rápidos y facilite cambios según las nuevas necesidades.

- **Simplicidad en el diseño:** Siguiendo el principio "diseñar para el día de hoy", el diseño aborda solo los requerimientos actuales, sin anticipar funcionalidades futuras no definidas. Todo el equipo participa en el proceso, incluidos desarrolladores, testers y el cliente.
- **Refactorización continua:** A lo largo del desarrollo, se realiza refactorización constante para mantener el diseño limpio, sencillo y fácil de mantener.
- **Entradas y salidas:**
 - **Entradas:** Las historias de usuario proporcionan el contexto funcional y los requerimientos específicos. Las prioridades del cliente guían el diseño, y las pruebas unitarias iniciales ayudan a establecer estructuras de código que cumplan con los criterios de aceptación.
 - **Salidas:** El diseño inicial establece una estructura simple, complementado con diagramas o bocetos que muestran cómo interactúan los componentes principales.
- **Responsables:**
 - **Developer:** Los desarrolladores son responsables del diseño técnico y de crear la estructura inicial del sistema, además de elaborar diagramas representativos.
 - **Coach:** Guía al equipo para que el diseño siga los principios de XP y esté alineado con las necesidades del cliente.
 - **Tracker:** Monitorea el progreso del diseño, asegurándose de que se cumpla el tiempo estimado y recopilando métricas de eficacia.
 - **Tester:** Participa indirectamente al definir los casos de prueba funcionales, ayudando a orientar el diseño hacia soluciones que cumplan los criterios de

Gestión de Acciones	Versión: 1.0
Documento de Flujo de Trabajo	Fecha: 20/11/2024

aceptación.

3.3.3 **Codificación**

El equipo de desarrollo transforma las historias de usuario y el diseño inicial en código funcional que cumpla con los requerimientos del cliente. Esta etapa se caracteriza por un enfoque colaborativo, donde se emplean prácticas clave de XP para garantizar la calidad y la agilidad del software.

Una de las prácticas centrales es la programación en pareja, en la cual dos desarrolladores trabajan juntos en el mismo código: uno escribe mientras el otro revisa y sugiere mejoras en tiempo real.

La etapa de codificación también enfatiza la **simplicidad y la refactorización continua**.

- **Entradas y Salidas:**

- **Entradas:** Como entradas están las historias de usuario definen los requerimientos funcionales que el código debe cumplir, sirviendo como guía para la implementación. El diseño inicial proporciona la estructura básica y los componentes del sistema que se deben desarrollar. Las pruebas unitarias previas (TDD) guían la escritura del código, asegurando que se cubran los criterios de aceptación.
- **Salidas:** El principal resultado de esta etapa es el código funcional, que implementa las funcionalidades definidas en las historias de usuario. Además, se obtienen pruebas unitarias aprobadas, ya que las pruebas definidas previamente se ejecutan para validar que el código cumpla con los requisitos y no introduzca errores. A medida que avanza el desarrollo, el código refactorizado mejora en legibilidad, eficiencia y mantenibilidad sin alterar su funcionalidad.

- **Responsables:**

- **Developer:** El desarrollador es el principal responsable de escribir el código funcional. Utiliza las historias de usuario y el diseño inicial como guía para implementar las funcionalidades del sistema.
- **Coach:** El coach proporciona orientación técnica y metodológica al equipo. En la fase de codificación, asegura que el equipo siga los principios de XP, como la simplicidad del código, la programación en pareja y el uso adecuado de las pruebas unitarias.
- **Tracker:** El tracker monitorea el progreso del equipo durante la codificación. Se asegura de que las tareas se completen dentro de los plazos establecidos y registra el avance del proyecto.
- **Tester:** El tester desempeña un papel clave en esta fase al diseñar y ejecutar las pruebas unitarias y de integración para validar que el código cumpla con los criterios de aceptación.

3.3.4 **Pruebas**

El principal objetivo es garantizar que el software cumpla con los requisitos definidos en las historias de usuario y que se entregue un producto funcional y sin errores. Esta fase está estrechamente vinculada con las etapas anteriores de desarrollo, ya que las pruebas se basan en el código escrito y las historias de usuario implementadas.

- **Entradas y Salidas:**

- **Entradas:** Como entradas está el código funcional implementado que se somete a pruebas para verificar su correcto funcionamiento, guiado por las historias de usuario completadas que aseguran el cumplimiento de los requisitos. Las pruebas unitarias (TDD) verifican el funcionamiento de cada componente, y los criterios

Gestión de Acciones	Versión: 1.0
Documento de Flujo de Trabajo	Fecha: 20/11/2024

de aceptación validan que el sistema cumpla con las expectativas del cliente. Todo esto se organiza a través del plan de pruebas.

- **Salidas:** Las salidas de la fase de pruebas incluyen la identificación de errores o defectos en el sistema, que serán reportados para su corrección. También se tiene la validación de pruebas aprobadas, lo que confirma que el código cumple con los criterios de aceptación y pasa las pruebas unitarias e integrales. Además, pueden surgir mejoras o refactorizaciones del código para solucionar problemas detectados y optimizar el software.
- **Responsables:**
 - **Developer:** El desarrollador se asegura de que el código escrito cumpla con los criterios de aceptación y esté listo para ser probado.
 - **Coach:** El coach guía al equipo en la práctica de pruebas y asegura que se sigan las mejores prácticas en el proceso de desarrollo, incluidas las pruebas.
 - **Tracker:** El tracker se encarga de monitorear el progreso de las pruebas, asegurándose de que se realicen de acuerdo con el plan.
 - **Tester:** El tester es responsable de ejecutar las pruebas definidas, ya sean unitarias, de integración o funcionales. Verifica que el código funcione según lo esperado y que cumpla con los criterios de aceptación.

3.4 Roles y Responsabilidades

En el marco de Extreme Programming (XP), el enfoque es profundamente colaborativo, lo que significa que todos los miembros del equipo participan activamente en cada una de las etapas del desarrollo. Esto implica que, independientemente del rol de cada integrante, todos trabajarán de manera conjunta para alcanzar el objetivo común y entregar un producto de valor.

De esta manera, las decisiones, la ejecución de tareas, la validación y la mejora continua serán llevadas a cabo tanto de forma individual como colectiva. Cada miembro contribuirá con su experiencia y habilidades, y se apoyarán mutuamente a lo largo de todo el proceso, fomentando un ambiente de trabajo en equipo y comunicación constante para garantizar el éxito del proyecto.

3.5 Herramientas y Recursos

- **Gestión de tareas:** Trello.
- **Desarrollo:** IDEs como Visual Studio Code.
- **Control de versiones:** Git.
- **Pruebas:** SonarQube.

3.6 Iteración o Ciclo

Para el desarrollo de este proyecto se utilizará como base el marco de Extreme Programming (XP), por lo cual las fases que se han mencionado en las etapas de flujo se repetirán de manera ordenada hasta conseguir el proyecto, como lo indica la Imagen 2.

Gestión de Acciones	Versión: 1.0
Documento de Flujo de Trabajo	Fecha: 20/11/2024

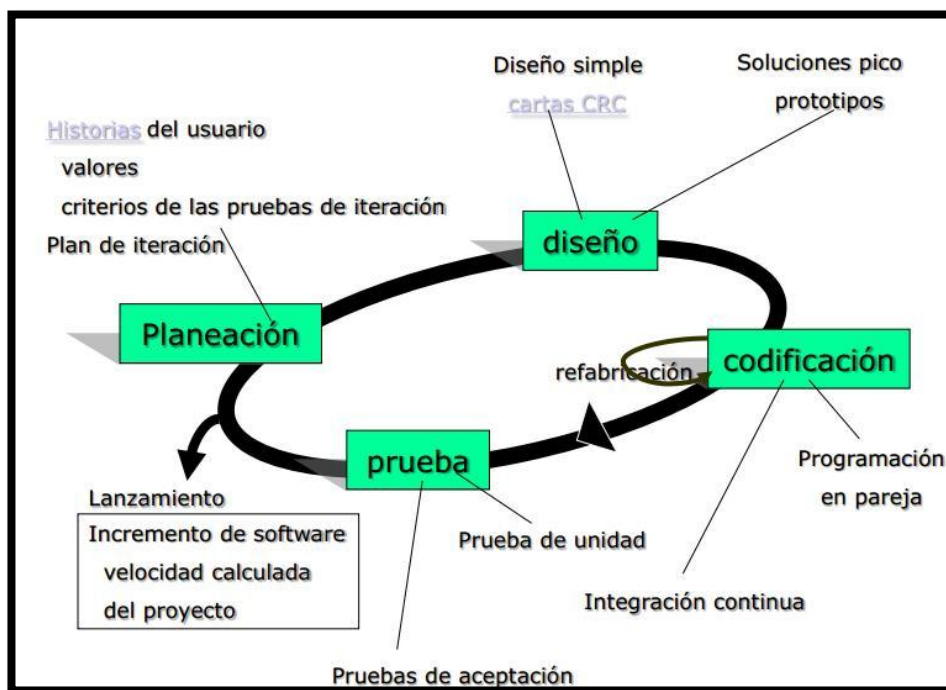


Imagen 2. Iteraciones del Proyecto

3.7 Dinámica del Equipo:

- **Colaboración Remota con Encuentros Ocasionales**
El equipo trabajará principalmente de manera remota, utilizando herramientas de comunicación y colaboración en línea para mantener una conexión constante entre todos los miembros. Cuando las circunstancias lo permitan, se organizarán reuniones presenciales o virtuales para reforzar el trabajo en conjunto y fomentar una colaboración más estrecha.
- **Revisión Constante**
Cada integrante del proyecto realizará en su tiempo disponible una revisión constante del proyecto para evaluar el avance del proyecto, posibles problemas y notificará al grupo de manera rápida para tomar medidas entre todos