

Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ciencias y Sistemas  
Análisis y Diseño de Sistemas 1  
Sección N  
Ing. Ricardo Morales  
Aux. Raul Mendizabal

# GiftHub

## Objetivos

- Que el estudiante entienda y aplique el concepto de integración continua.
- El estudiante entenderá las funcionalidades generales de un servidor de integración continua.
- El estudiante aprenderá a utilizar la herramienta Jenkins como servidor de integración continua.
- El estudiante entenderá como las pruebas unitarias y funcionales ayudan a demostrar la calidad de software.
- El estudiante aprenderá a comunicarse y presentar avances al product owner.

## Descripción General

Esta práctica consiste en tener un proceso automatizado que, después que cada desarrollador suba código al repositorio, se obtenga la última versión, se compile, se ejecuten pruebas funcionales y se coloquen los resultados en una ubicación determinada.

El sistema GiftHub necesita una herramienta de integración continua, la cual será Jenkins. Esta herramienta ayudará a compilar el código fuente y también correr tests para así obtener un ejecutable exitoso (BUILD SUCCESSFUL). **Se deberá utilizar Jenkins de modo que cada vez que se realice un commit se corran las tareas necesarias.**

Se usará un sistema de control de versiones distribuido para almacenar el código de la aplicación utilizando el software Git. El host de este servidor será Github. El sistema GiftHub debe estar integrado, es decir Github y Jenkins por lo que se espera que se envíen correos automáticamente a los cambios realizados en el sistema de control de versiones.

Se debe realizar el código de las funcionalidades del sistema GiftHub, este debe compilar y ejecutar las pruebas de cada funcionalidad. La herramienta de integración continua notificará a los desarrolladores acerca del resultado de estos pasos ya sea si fue exitoso o no.

## Información de Negocio

Gifthub es una empresa que se dedica a la venta centralizada de tarjetas de regalo electrónicas de plataformas como Google, PlayStation, Amazon, Microsoft etc... Por lo que se posee un contrato con las respectivas entidades para vender sus productos en la plataforma de Gifthub

Gifthub ya posee una API interna de su catálogo el cual consiste de 3 recursos:

<https://my-json-server.typicode.com/CoffeePaw/AyD1API>

- /Card
  - Id: Identificador de GiftCard.
  - name: Nombre de la plataforma a la que pertenece la GiftCard.
  - image: Imagen proveída por la plataforma.
  - chargeRate: El recargo que cobra # por venta de GiftCard.
  - Active: Bandera que indica si la plataforma está habilitada
  - availability: Lista de referencias a código de valores disponibles para la GiftCard.
- /Value
  - Id: Identificador de valor.
  - Total : Valor asociado al identificador.
- /TasaCambio
  -

A Gifthub le interesa saber los siguientes datos para los usuarios dentro de la plataforma:

- Username
- Correo
- Contraseña
- Nombres
- Apellidos
- DPI
- Edad
- Inventario de Tarjetas compradas u obtenidas
- Transacciones de tarjetas

Github tiene el derecho de generar sus propias tarjetas de regalo de las plataformas con las que se tiene acuerdo. Por lo que a la hora de generar una tarjeta de regalo debe tomar en cuenta lo siguiente:

- Cada GiftCard está asociada al usuario que la compro
- Cada GiftCard será identificada por un código alfanumérico random de 8 caracteres generado a la hora de crear la GiftCard.

Por seguridad de la empresa y del cliente Gifthub no guardará el número de la Tarjeta de Crédito sino lo encriptará con el siguiente formato **####XXXXXXXX####**, Este dato debe ser almacenado.

## Funcionalidades Requeridas

Dado a las necesidades de Gifthub se solicita que se realice las siguientes funcionalidades:

### Login

Los usuarios registrados podrán acceder a su cuenta antes de tener acceso a las funcionalidades de GiftCards con las siguientes credenciales

- Email o Username
- Password

### Registro

Los usuarios deberán registrarse en la plataforma para tener acceso a los módulos de Gifthub. Uno por correo y username único.

### Compra de GiftCards

Los usuarios logeados podrán ver el catálogo de GiftCards disponibles, el cual solo muestra GiftCards activas y poder realizar la compra de cualquiera de estas GiftCards. Una vez seleccionadas las GiftCards y las cantidades unitarias el usuario podrá realizar el pago de GiftCards(Ver la funcionalidad PAGO DE TARJETA). De ser aprobado se generará una GiftCards de regalo el cual estará en el inventario del usuario. Si la compra fue exitosa o no, se deberá guardar historial de la transacción.

### Pago de Tarjeta

Módulo en donde el usuario ingresará los siguientes campos:

- Número de tarjeta de crédito (Integer de 16 caracteres)
- Nombre en la tarjeta (String)
- Fecha de expiración (MM/YY)
- Código verificador (Integer de 3 caracteres)
- Monto a pagar **\*Obtenido de la selección de compra del usuario**
- Moneda con la cual se va a pagar **\*Se debe utilizar la api para obtener el cambio de la moneda**

Al enviar la transacción se deben de realizar las validaciones correspondientes.

## Historial de Compra

Apartado en el cual el usuario podrá ver todo el historial de compras exitosas o fallidas de GiftCards.

## Inventario de GiftCards

Una vez compradas las GiftCards el usuario podrá ver sus GiftCards compradas en una lista.

## Perfil de Usuario

El usuario podrá ver y editar su perfil.

## Regalo de GiftCards

El usuario tendrá la opción de regalar GiftCards que ya tenga en su inventario a otro usuario del sistema.

## Usuario Administrador

El usuario administrador podrá ver el detalle de transacciones de las GiftCards de regalo de todos los usuarios y también tiene que tener habilitada la opción de observar el catálogo.

# Control de Calidad

Para seguir los altos estándares de Gifthub se requiere asegurar la calidad del software a entregar por medio de los siguientes puntos.

## Pruebas unitarias

Cada función que se hace en el sistema debe pasar por una prueba unitaria, para verificar el correcto funcionamiento de la unidad de código. Las herramientas a utilizar dependen del lenguaje de programación. **WEB**

## Pruebas funcionales

Cada funcionalidad que se realice en el sistema debe pasar por sus respectivas pruebas funcionales, se plantearán **casos de pruebas** funcionales, las herramientas a utilizar quedan a discreción.

## Sonar Qube

Se implementará esta herramienta para la evaluación de métricas del software, en esta categoría están las siguientes métricas de las cuales solo deben escoger 3 para ser evaluadas durante la calificación

- Complejidad ciclomática
- Código duplicado
- Comentarios

- Test
- Violaciones

## Manual Tecnico

El manual técnico deberá contener una descripción de la arquitectura empleada en el proyecto, incluyendo diagrama de base de datos y de cada uno de los componentes utilizados y los módulos realizados. Así como una guía de instalación y configuración de Jenkins y Sonar Qube así como una relación entre las pruebas realizadas con el código evaluado.

## Manual de Usuario

El manual de usuario deberá ser una guía de utilización de su página web

## SCRUM

Por propósitos de auditoría Github solicita que se trabaje con la metodología SCRUM por lo que se requiere la utilización de un tablero.

El product Owner tendrá breves reuniones con el equipo de trabajo cada sábado de 10:30 a 12:30, en donde el equipo de trabajo presentará sus avances y obtener información del proyecto. *(Lo cual tendrá ponderación en la calificación)*

## Fecha de Entrega Fase 1

17/10/2020

Aspectos a Evaluar:

- Pruebas Unitarias
  - Mocks
- Manual Tecnico
- Manual de Usuario
- Arquitectura
- Sonar Qube
- Gitflow
- Scrum

## Fecha de Entrega Fase 2

31/10/2020

Aspectos a Evaluar:

- Pruebas Funcionales
- Aplicación Funcional
- Integración Continua
- Documentación Complementaria Fase 1(Casos de Pruebas, Integración continua)
- Gitflow
- Scrum