

Evidencia: GA4-220501095-AA1-EV01.

Informe Del Modelo Conceptual Orientado A Objetos

Aprendiz: Juan Carlos Lopez Moreno

Instructor: Alvaro Esteban Betancourt Matoma

Área Técnica

Servicio Nacional de Aprendizaje – SENA

Técnico en Programación de Aplicaciones y Servicios Para la Nube

Código: 3186263

Noviembre 2025

Introducción

Café Yamara es una empresa dedicada a la producción de café orgánico en grano y molido localizada en el municipio de Guadalupe, Santander Colombia. Tras un primer contacto se logró identificar previamente como se determinaban los requisitos del software a construir, se logró a identificar los stakeholders y como estos interactuaban en la lógica de negocio, así nos ayudó a determinar tras diversos métodos de recolección de datos cada uno de los requerimientos.

En este informe se realiza una descripción de las clases identificadas mediante la recolección y el análisis de los requisitos del software a construir en el módulo 1. Mediante dicho documento y tras haber hecho un análisis profundo con las partes interesadas se logró determinar todos los requisitos del software a construir, mediante este se pudo determinar las clases, sus atributos y por último los métodos que estas clases pueden tener.

Para determinar esto se hizo una lectura del documento delicitación de requisitos del software, se identificaron las clases a construir mediante la lista de requerimientos funcionales, los atributos se identificaron dependiendo de la naturaleza de cada una de las clases y de la perspectiva del producto, los métodos se retomaron del apartado de funciones del producto, en donde se determinaron anteriormente cada una de las funcionalidades de cada una de las clases.

Así mismo todo esto se reforzó siguiendo los datos obtenidos en la historia de usuario que se encuentra dentro del documento delicitación de requisitos, esto nos permite entender mejor como es cada una de las interacciones dentro de la plataforma web, permitiéndonos determinar cada uno de los atributos y también de los métodos.

Alcance

El siguiente documento tiene como alcance determinar cada una de las clases a realizar incluyendo sus atributos; es decir cada una de las características que pueden ser físicas o no físicas dependiendo de la clase, sus métodos; que se refieren a cada una de las funcionalidades que puede hacer la clase o sus herencias dentro de la plataforma, todo esto enmarcado en el informe técnico IEEE 830.

El informe se limita a incluir solo las clases con sus atributos y métodos, sin incluir aún las herencias, las relaciones entre cada una de las clases ni tampoco la implementación mediante código fuente que incluye la elección del lenguaje de programación a elegir según las necesidades de la implementación que se piensa realizar, la base de datos y la estructura de datos que se deseará emplear así como el motor de base de datos y el tipo de base de datos a utilizar (relacional o no relacional).

Lista de Clases Identificadas

1. Clase Usuario

Representa a todas las personas que interactúan con la plataforma, ya sean visitantes, clientes o administradores. Se decidió incluirla porque los requisitos funcionales RF06 y las historias de usuario HU-06 y HU-07 mencionan explícitamente la necesidad de registro, inicio de sesión, gestión de roles y autenticación. Sus métodos permiten realizar las operaciones básicas de gestión de usuarios (crear cuenta, iniciar sesión, asignar rol, bloquear, etc.).

2. Clase Administrador

Deriva conceptualmente del usuario, pero con privilegios especiales para gestionar contenido y productos. Se decidió crearla porque los requisitos describen tareas exclusivas del administrador, como la gestión de productos e imágenes (RF04, HU-07) y la administración

general de la plataforma (HU-08). Sus métodos permiten crear, modificar y eliminar productos, así como administrar usuarios.

3. Clase Producto

Modela los artículos que la finca ofrece a través de la plataforma (café u otros derivados). Se decidió incluirla ya que los requisitos RF04 (listar productos) y RF05 (gestionar ventas) giran en torno a su visualización y compra. Sus métodos implementan las operaciones CRUD (crear, leer, actualizar y eliminar), necesarias tanto para el catálogo público como para la administración interna.

4. Clase CarritoCompra

Representa el contenedor temporal donde el cliente agrupa los productos antes de realizar la compra. Se incluye porque el RF05 especifica un flujo de compra con carrito, confirmación y pago. Sus métodos permiten agregar o eliminar productos, calcular el total y vaciar el carrito después del pago.

5. Clase Venta

Registra cada transacción realizada por los clientes dentro del sistema. Se decidió incluirla ya que el sistema debe generar ventas, procesar pagos e identificar el número de orden (RF05, HU-05). Sus métodos permiten registrar, confirmar y consultar las ventas, así como enviar la notificación de confirmación al cliente.

6. Clase Pago

Representa el proceso de integración con la pasarela de pago (Stripe u otra). Su creación se justifica por la necesidad de separar la lógica de pago de la lógica de venta, según RF05 y los requisitos no funcionales de seguridad y confiabilidad (RNF-03). Sus métodos controlan el inicio del pago, su validación, registro y verificación del estado final.

7. Clase Finca

Modela la información institucional y visual de la finca Miraflores. Se decidió incluirla por los requisitos RF01 y RF03, donde se solicita mostrar la historia, filosofía, ubicación e imágenes de la finca. Esta clase centraliza los datos que se mostrarán en el módulo “Acerca de nosotros” y en la sección de ubicación.

8. Clase PasoPreparacion

Representa cada paso del proceso para preparar una taza de café. Se incluye porque RF02 detalla la necesidad de mostrar pasos con texto e imágenes en una interfaz interactiva. Sus métodos permiten listar y acceder a los pasos en orden, garantizando una experiencia guiada para el usuario visitante.

9. Clase Galeria

Contiene las imágenes relacionadas con la finca, los productos o las instalaciones. Se decidió incluirla porque RF03 exige mostrar una galería de fotos, con miniaturas y vistas ampliadas. Sus métodos permiten agregar, eliminar y listar imágenes en el sistema, especialmente para que el administrador gestione el contenido visual.

10. Clase Mapa

Representa la integración con servicios externos (como Google Maps u OpenStreetMap) para mostrar la ubicación de la finca. Su inclusión se justifica por el requisito RF03, que menciona un mapa interactivo centrado en Guadalupe, Santander. Sus métodos permiten mostrar, centrar o actualizar la ubicación en el mapa.

11. Clase Autenticación

Gestiona la seguridad del sistema mediante tokens JWT y control de sesiones. Se decidió crearla para responder a los requisitos no funcionales de seguridad (RNF-03) y al RF06 (gestión

de usuarios y roles). Sus métodos manejan la generación, validación y expiración de los tokens, así como el bloqueo por intentos fallidos.

Usuario	Producto	CarritoCompra
<ul style="list-style-type: none"> + id_usuario + nombre + apellido + correo + contraseña + rol + estado_cuenta + fecha_registro <ul style="list-style-type: none"> + registrarUsuario() + iniciarSesion() + cerrarSesion() + recuperarContraseña() + verificarEmail() + asignarRol() + bloquearUsuario() 	<ul style="list-style-type: none"> + id_producto + nombre + descripción + precio + imagen + stock + categoria + fecha_creación <ul style="list-style-type: none"> + crearProducto() + leerProducto() + actualizarProducto() + eliminarProducto() + listarProductos() + buscarProducto() 	<ul style="list-style-type: none"> + id_carrito + id_usuario + listaProductos + total + fecha_creación <ul style="list-style-type: none"> + agregarProducto() + eliminarProducto() + calcularTotal() + vaciarCarrito() + obtenerCarritoPorUsuario()
Venta	Pago	Finca
<ul style="list-style-type: none"> + id_venta + id_usuario + id_carrito + total + fecha_venta + estado + numero <ul style="list-style-type: none"> + registrarVenta() + procesarPago() + confirmarVenta() + enviarEmailConfirmacion() + obtenerHistorialVentas() 	<ul style="list-style-type: none"> + id_pago + id_venta + metodo_pago + monto + estado + fecha_pago + token_transacción <ul style="list-style-type: none"> + iniciarPago() + registrarPago() + verificarEstado() + validarPago() 	<ul style="list-style-type: none"> + id_finca + nombre + historia + filosofía + ubicación + imagenes[*] <ul style="list-style-type: none"> + mostrarInformación() + listarImagenes() + obtenerUbicación()
PasoPreparacion	Galeria	Mapa
<ul style="list-style-type: none"> + id_paso + titulo + descripcion + imagen + orden <ul style="list-style-type: none"> + listarPasos() + obtenerPasoPorOrden() 	<ul style="list-style-type: none"> + id_imagen + titulo + url + descripción + fecha_subida <ul style="list-style-type: none"> + agregarImagen() + eliminarImagen() + listarGaleria() 	<ul style="list-style-type: none"> + id_map + api_key + latitud + longitud + zoom <ul style="list-style-type: none"> + mostrarMap() + centrarMapa() + actualizarUbicación()
Autenticación	Administrador	
<ul style="list-style-type: none"> + tokenJWT + fecha_expiracion + intentos_fallidos <ul style="list-style-type: none"> + generarToken() + validarToken() + excluirSesion() + bloquearPorIntentos() 	<ul style="list-style-type: none"> + id_admin + contraseña + nombre + correo + permisos[*] <ul style="list-style-type: none"> + crearProducto() + modificarProducto() + eliminarProducto() + gestionarUsuarios() + generarReporte() 	