

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**MANUAL TÉCNICO:**  
COFFEE LINK



**Semestre:** 2025-1

Índice

1. Cronograma de actividades	1
1.1. Diagrama de Gantt . . . . .	2
2. Estructura del programa	3
3. Tecnologías usadas	5
4. Procedimientos	6
5. Referencias	7

## 1 Cronograma de actividades

### Fase 1: Planificación y Diseño

- Inicio del diseño de la propuesta del proyecto y desarrollo del diseño de UI y modelos 3D (2024-10-13 - 2024-11-08):
  - Se elaboró la idea inicial del proyecto, estableciendo los objetivos y alcance.
  - Desarrollo de los primeros diseños de la interfaz de usuario (UI) y creación de modelos 3D necesarios para la aplicación.
- Definición detallada de funcionalidades del MVP (2024-10-21 - 2024-11-15):
  - Se definieron las funcionalidades esenciales para el producto mínimo viable (MVP), asegurando que se cubran las necesidades básicas del usuario.
- Diseño de la interfaz de usuario (UI) y experiencia de usuario (UX) (2024-10-24 - 2024-10-28):
  - Diseño detallado de la UI, considerando la experiencia del usuario (UX) para garantizar una navegación intuitiva y efectiva.
- Creación de wireframes y prototipos (2024-10-29 - 2024-11-04):
  - Elaboración de wireframes y prototipos interactivos para visualizar y validar el diseño con los interesados antes de la implementación.

### Fase 2: Desarrollo

- Configuración del proyecto en Unity (2024-11-05 - 2024-11-08):
  - Se realizó la configuración inicial del proyecto en Unity, estableciendo la estructura y dependencias necesarias.
- Implementación del menú de navegación, flujo de UI, configuración del tema, y control de UI (2024-11-09 - 2024-11-17):
  - Desarrollo e integración del menú de navegación.
  - Implementación del flujo de la interfaz de usuario (UI).
  - Configuración del tema visual del proyecto.
  - Control y refinamiento de elementos de UI para asegurar su funcionamiento correcto.
- Actualización de .gitignore (2024-11-09):

[illegible]

## 2 Estructura del programa

Las clases del proyecto están interconectadas para proporcionar un funcionamiento cohesivo del sistema. La clase **ARLogic** se encarga de la lógica de la interfaz de usuario para la realidad aumentada (AR), gestionando componentes como etiquetas, botones y contenedores visuales. **MainMenuEvents** inicializa estos componentes y maneja eventos como el inicio de sesión, mientras que **NavigationMenuEvents** utiliza una instancia de **ARLogic** para la navegación dentro de la aplicación, configurando eventos de clic y manejando secciones de la UI. **CoffeSO**, que hereda de **ScriptableObject**, centraliza datos del café utilizados por **CoffeTarget**, que hereda de **MonoBehaviour**, y controla la lógica y presentación de objetivos relacionados con el café en el sistema. En conjunto, estas clases interactúan para ofrecer una experiencia de usuario integrada, utilizando los datos y lógica de **CoffeSO** y **CoffeTarget**, junto con la gestión de eventos y navegación proporcionada por **MainMenuEvents** y **NavigationMenuEvents**. Así, se logra una interfaz de usuario funcional y fluida dentro del entorno de realidad aumentada del proyecto. A continuación se detallan las clases y su funcionamiento.

### Clase **ARLogic**

- Espacios de nombres importados:

- `System.Collections`
- `System.Collections.Generic`
- `UnityEngine`
- `UnityEngine.UIElements`

- Definición de la clase **ARLogic**:

- La clase contiene varios campos privados que son componentes de la interfaz de usuario de Unity:
  - `Label label`
  - `Button nextButton`
  - `Button previousButton`
  - `Button registerButton`
  - `VisualElement mapContainer`
  - `Label auxLabel`
  - `Label registerLabel`
  - `RadioButtonGroup registerGroup`

### Clase **MainMenuEvents**

- Métodos:

- **Awake:**
  - Inicializa varios componentes de la interfaz de usuario obtenidos a través de un componente `UIDocument`.

- Componentes: `_loginButton`, `_usernameField`, `_passwordField`.
- Registra callbacks para eventos de enfoque y de clic en el botón de inicio de sesión.
- **OnLoginButtonClicked:**
  - Obtiene los valores ingresados en los campos de nombre de usuario y contraseña.

## Clase `NavigationMenuEvents`

### ■ Componentes:

- Instancia de `ARLogic`.
- Propiedad de estilo personalizada estática para un color secundario.

### ■ Métodos:

#### • **Start:**

- Inicializa varios componentes de la interfaz de usuario obtenidos a través de un componente `UIDocument`.
- Componentes: `arContainer`, `databaseContainer`, `discoverContainer`, `settingsContainer`, `mainContainer`, `registerContainer`, `databaseScrollContainer`.
- Registra un callback para el evento `CustomStyleResolvedEvent`.
- Configura un evento de clic para `DatabaseButton` que llama al método `ShowDatabase`.

## Clase `CoffeSO`

### ■ Atributos:

- `coffeName`: Almacena el nombre del café.
- `description`: Almacena una descripción del café.
- Sprite de mapa asociado (no mostrado completo en el código visible).

### ■ Características:

- Hereda de `ScriptableObject`.
- Marcada con el atributo `CreateAssetMenu`.

## Clase `CoffeTarget`

### ■ Atributos:

- Referencia a un `CoffeSO`.
- Referencias a prefabs y objetos de la escena: `starsPrefab`, `starsParent`, `beanParent`.

- Arreglo de transformaciones: `starsPositions`.
- Materiales: `starsEnabled`, `starsDisabled`.

■ **Características:**

- Hereda de `MonoBehaviour`.
- Maneja la lógica y la presentación visual de un objetivo relacionado con el café en el sistema.

### 3 Tecnologías usadas

■ **Blender:**

- Blender es una herramienta de código abierto y gratuita para el modelado 3D, la animación, la renderización y más.
- Utilizamos Blender para crear modelos tridimensionales de los objetos que forman parte de la aplicación, asegurando que estos modelos sean detallados y de alta calidad.
- Su amplia gama de características y compatibilidad con otros programas lo hace ideal para integrar modelos 3D en Unity.

■ **Unity:**

- Unity es un motor de juego muy popular que proporciona un entorno de desarrollo integrado para la creación de videojuegos y aplicaciones interactivas.
- Se utilizó como la plataforma principal para desarrollar nuestro proyecto, proporcionando las herramientas necesarias para implementar la lógica del juego, el manejo de eventos y la interacción con el usuario.
- Sus capacidades avanzadas permiten el uso de scripts en C#, la integración con múltiples tecnologías y la creación de experiencias inmersivas en realidad aumentada.

■ **Vuforia:**

- Vuforia es una plataforma líder en la industria para aplicaciones de realidad aumentada (AR).
- Empleamos Vuforia para manejar la detección y el seguimiento de imágenes y objetos en el entorno de AR, lo que permite una interacción precisa y en tiempo real con elementos del mundo real.
- Su integración con Unity facilita la implementación de experiencias AR avanzadas sin necesidad de conocimientos profundos en programación.

■ **UI Toolkit:**

- UI Toolkit es un conjunto de herramientas de Unity diseñado para la creación y desarrollo de interfaces de usuario.

- Utilizamos UI Toolkit para diseñar y personalizar la interfaz de usuario de nuestra aplicación, asegurando que sea intuitiva y fácil de usar.
- Proporciona una manera eficiente de gestionar la lógica de la UI, permitiendo una mayor flexibilidad y control sobre los elementos visuales y su comportamiento.

## 4 Procedimientos

Para configurar el proyecto en Unity, el primer paso es descargar e instalar la versión más reciente de Unity desde su sitio oficial. Una vez instalada, se abre Unity Hub y se crea un nuevo proyecto seleccionando la plantilla adecuada, en este caso, la plantilla 3D debido a que se usará Vuforia para la realidad aumentada. Una vez creado el proyecto, se configuran los ajustes básicos, como el nombre, la organización, y la plataforma de destino. Posteriormente, se importan todos los paquetes necesarios a través del gestor de paquetes de Unity, con la excepción de Vuforia, que debe descargarse desde su página oficial e importarse manualmente. La organización de la estructura del proyecto se lleva a cabo creando carpetas específicas para scripts, prefabs, escenas y recursos, asegurando que todos los elementos estén categorizados y sean fácilmente accesibles.

Para integrar Vuforia en el proyecto, primero se deben descargar los paquetes desde la página oficial de Vuforia. Luego, en Unity Hub, se carga el proyecto y se abre el Package Manager desde el menú "Window". En el Package Manager, se selecciona la opción "Add package from disk" se navega hasta el archivo descargado de Vuforia. A continuación, se añade un "Vuforia AR Camera" desde el menú de GameObject y se configuran las licencias y targets en el panel de configuración de Vuforia, lo que permite al proyecto utilizar las capacidades de AR proporcionadas por Vuforia.

El modelado 3D de objetos se realiza utilizando Blender, una herramienta de código abierto que permite la creación, escultura y texturizado de modelos tridimensionales. Una vez que los modelos están completos, se aseguran de estar bien mapeados y texturizados adecuadamente. Los modelos se exportan desde Blender en un formato compatible con Unity, como .fbx, y luego se importan a Unity arrastrándolos a la carpeta de Assets. En Unity, se ajustan las escalas y los materiales para que coincidan con las especificaciones del proyecto, preparando los modelos para ser utilizados como prefabs o directamente en las escenas del proyecto.

El diseño de la interfaz de usuario se lleva a cabo utilizando UI Toolkit de Unity. Primero, se asegura que el paquete de UI Toolkit esté instalado desde el Package Manager. Luego, se crea un nuevo UIDocument en la escena, que actúa como contenedor de la UI. Desde el panel de UI Builder, se diseña y estructura la interfaz de usuario utilizando elementos visuales como botones, etiquetas y contenedores. Estos elementos se arrastran y sueltan desde el panel de biblioteca al canvas del UI Builder, y se personalizan sus propiedades y estilos utilizando el inspector. Una vez diseñada la UI, se utilizan scripts en C# para agregar lógica y funcionalidad a los elementos interactivos, asegurando una experiencia de usuario coherente y efectiva.

Finalmente, para utilizar el proyecto descargándolo del repositorio, se debe clonar el repositorio desde la plataforma de control de versiones, como GitHub, utilizando un



comando como ‘git clone’. Después de descargar el proyecto, se abre Unity Hub y se añade el proyecto clonado. Es necesario obtener una licencia de Vuforia desde su página oficial y configurarla en el asset creado en la carpeta ‘Resources’ del proyecto. Al abrir el proyecto en Unity, se debe verificar que todos los paquetes necesarios estén correctamente instalados y configurados, incluyendo la importación de cualquier paquete adicional mencionado en la documentación del proyecto.

## 5 Referencias

- Blender. (s.f.). *Blender.Org - Home of the Blender Project - Free and Open 3D Creation Software*. Consultado el 18 de noviembre de 2024, desde <https://www.blender.org/>
- GitHub. (s.f.). *GitHub · Build and Ship Software on a Single, Collaborative Platform · GitHub*. Consultado el 18 de noviembre de 2024, desde <https://github.com/>
- Unity. (s.f.). *Home — Engine Developer Portal*. Consultado el 18 de noviembre de 2024, desde <https://developer.vuforia.com/home>
- Vuforia. (s.f.). *Plataforma de Desarrollo En Tiempo Real de Unity — Motor de 3D, 2D, VR y AR*. Consultado el 18 de noviembre de 2024, desde <https://unity.com/es>