

## Orientación para el II Proyecto: EdgeAI - **Sistema embebido para Reconocimiento de Terreno**

TIPO DE SESIÓN: ASINCRÓNICA

### **Objetivo de aprendizaje**

Desarrollar en el educando, las habilidades y destrezas necesarias para el análisis, síntesis y creación de sistemas embebidos de pequeña a mediana escala que pudieran ser requeridos en el contexto de los proyectos de su especialidad.

### **Contenidos**

Tema 4: Herramientas de hardware y software para el diseño de sistemas embebidos.

- Kit de desarrollo
- Tool Chain
- Depuradores
- Simuladores
- Sintetizadores de sistemas operativos
- Estaciones de trabajo

Tema 6: Programación de Sistemas Embebidos.

Tema 7: Diseño de sistemas Embebidos

Tema 8: Depuración, verificación y validación de

### **Persona facilitadora**

Dr. Ing. Johan Carvajal  
Godínez  
[johcarvajal@itcr.ac.cr](mailto:johcarvajal@itcr.ac.cr)  
Consulta: L, M 11 am-12 pm

### **PRESENTACIÓN**

Los algoritmos de clasificación de objetos son esenciales para el reconocimiento de terreno en aplicaciones que requieren procesamiento en tiempo real, como vehículos autónomos, drones, o robots de exploración. Estos algoritmos permiten identificar y categorizar características del terreno, como rocas, caminos, vegetación u obstáculos, lo cual es fundamental para la navegación segura y eficiente en entornos desconocidos o cambiantes. Sin embargo, el procesamiento de grandes volúmenes de datos en la nube introduce latencias que pueden ser inaceptables en aplicaciones críticas. Por ello, la necesidad de ejecutar estos algoritmos localmente, en dispositivos de borde (EdgeAI), se ha vuelto prioritaria, ya que esto reduce el tiempo de respuesta y mejora la eficiencia.

La computación de borde (EdgeAI) permite que los dispositivos cercanos al lugar donde se recopilan los datos procesen la información sin depender de conexiones a servidores remotos. Esto es especialmente importante en situaciones donde la conectividad puede ser limitada o inestable, como en áreas rurales o en misiones de exploración espacial. Al ejecutar los algoritmos de clasificación de objetos en el borde, se reduce la latencia, se optimiza el uso de ancho de banda y se mejora la privacidad al no enviar grandes volúmenes de datos sensibles a la nube. Además, estos sistemas pueden reaccionar rápidamente ante cambios en el entorno, lo que es crucial para la toma de decisiones en tiempo real.

Como parte del desarrollo de este proyecto se deben alcanzar los siguientes objetivos específicos:

1. Desarrollar una arquitectura física del sistema embebido por medio de un análisis operacional del sistema propuesto, y la síntesis de una propuesta de diseño usando herramientas de diseño asistido por computadora y herramientas de ingeniería de sistemas complejos.
2. Sintetizar un conjunto de aplicaciones de software capaces de implementar las funcionalidades requeridas para el sistema embebido por medio del uso de herramientas de visión por computador y aprendizaje máquina, así como el flujo de trabajo de Yocto Project.
3. Demostrar la operación correcta del sistema embebido mediante una presentación del desarrollo, implementación y verificación a partir de casos de uso específicos para la aplicación propuesta corriendo sobre una Raspberry pi o una Jetson Nano.

### Fecha / Calendario de entrega

Propuesta de diseño – 11 de octubre de 2024 antes de las 12 pm.

Bitácoras de actividades individuales y repositorio de código – 28 de octubre de 2024 antes de 12 pm

Demostración final del proyecto el 28 de octubre de 2024 antes de 12 pm.

### Valor puntaje y porcentaje

Propuesta de diseño – 30 puntos – 10%

Bitácoras y repositorio de código – 8 puntos – 5%

Demostración Final del proyecto – 12 puntos – 10%

### RUTA DE APRENDIZAJE

Para el desarrollo de la asignación crearán tres equipos de trabajo con los integrantes asignados por parte del profesor lo cuales deberán:

- Identificar un modelo de aprendizaje automático que pueda utilizarse para la aplicación propuesta.
- Optimizar el modelo de aprendizaje automático por medio de casos de uso del sistema embebido.
- Investigar el flujo de trabajo con la biblioteca OpenCV para la implementación de los algoritmos de aprendizaje máquina en la plataforma seleccionada.
- Sintetizar los casos de uso y requerimientos del sistema.
- Sintetizar las aplicaciones requeridas para operar y administrar el sistema embebido propuesto.
- Identificar las dependencias de software a nivel de sistema operativo.
- Consolidar las recetas requeridas para incluir las dependencias en el flujo de trabajo de Yocto Project.
- Sintetizar una imagen de Linux para un Raspberry pi/Jetson nano con la aplicación seleccionada siguiendo el flujo de trabajo de con Yocto Project y tensorflow lite/tensor RT/ OpenCV DNN.
- Instalar la imagen generada en Raspberry pi/Jetson Nano.
- Preparar un documento que describa paso a paso el procedimiento para la generación e instalación de dicha imagen.
- Preparar una demostración de la aplicación, así como el flujo de síntesis del sistema operativo y descripción de la aplicación y las recetas generadas.
- Documentar el proceso seguido por medio de una bitácora individual de trabajo.
- Demostrar el sistema operando en campo

El equipo debe establecer roles que serán documentados por medio de la bitácora individual de trabajo. Estos roles incluyen: un/una director del proyecto, un/una líder técnico o arquitecto de sistemas, un/una investigadora y un/una auditora encargada de gestionar la calidad del proyecto.

La siguiente tabla resume las responsabilidades asociadas a cada rol:

Rol	Responsabilidades/Funciones
Director(a) de proyecto	<ul style="list-style-type: none"> <li>• Llamar a reuniones</li> <li>• Punto de contacto con el profesor</li> <li>• Liderar las discusiones técnicas/documentar acuerdos.</li> </ul>
Líder Técnico	<ul style="list-style-type: none"> <li>• Proponer arquitecturas para el sistema</li> <li>• Implementar integración de componentes</li> <li>• Gestión de requerimientos del sistema</li> </ul>
Investigador (a)	<ul style="list-style-type: none"> <li>• Proponer soluciones a problemas encontrados en el diseño.</li> <li>• Proponer mejoras al software del sistema</li> </ul>

### Retroalimentación para el desarrollo de entregables.

	<ul style="list-style-type: none"> <li>• Documentar mejoras implementadas</li> </ul>
Auditor (a)	<ul style="list-style-type: none"> <li>• Proponer listas de cotejo para verificación</li> <li>• Revisión de documentos entregables</li> </ul>

Para el desarrollo de la propuesta de diseño, se pueden realizar consultas con el facilitador por medio del horario de consulta del curso.

Las sesiones de retroalimentación durante las consultas permitirán clarificar los alcances del proyecto, así como dar seguimiento al cumplimiento del cronograma de la propuesta de diseño.

Durante la revisión final se dará retroalimentación del nivel de logro de los criterios propuestos en las rúbricas de evaluación (ver instrumentos de evaluación).

### Formato de entrega y espacios

Ver los canales de MS teams para cada uno de los equipos de trabajo. Allí se colocarán instrucciones específicas para las entregas de los productos.

Los entregables se realizarán por medio de los canales privados de teams de cada uno de los grupos de trabajo. Allí crearán una carpeta para almacenar los documentos o enlaces necesarios.

1. La propuesta de diseño será por medio de un documento grupal en pdf (ver rúbrica para conocer la estructura del documento.) fecha de entrega 11 de octubre de 2024
2. Las bitácoras individuales serán por medio de un documento individual en formato pdf. Deben generar una bitácora por cada uno de los roles definidos en la ruta de aprendizaje. Fecha máxima de entrega el 28 de octubre de 2024. Las bitácoras deben incluir: fecha, actividades realizadas, productos obtenidos, problemas encontrados y referencias consultadas para su solución.
3. El repositorio público será por medio de la herramienta GitHub antes del 28 de octubre de 2024
4. La demostración final será realizada presencialmente en el edificio K1 el día 28 de octubre de 2024 a la hora asignada al grupo.

## Instrumento de evaluación

A continuación, se presentan rúbricas de evaluación propuestas para la evaluación de los entregables del proyecto. Se utilizó la técnica de rúbrica analítica.

## Evaluación de la propuesta de diseño

Item	Peso	3	2	1	0	Comentarios en el documento
		Lo esperado	Regular	Deficiente	Nulo	
1. Justificación del proyecto y revisión bibliográfica.	10%	La introducción refuerza la justificación del desarrollo del proyecto soportándose en 10 referencias adicionales a las expuestas en el instructivo.	La introducción amplía la justificación del desarrollo del proyecto soportado con ideas propias adicionales a las expuestas en el instructivo.	La introducción usa el mismo argumento presentado en el instructivo del proyecto sin agregar información que motive la importancia del desarrollo del mismo.	Le falta introducción en el documento de propuesta de diseño.	
2. Descripción y síntesis del problema.	10%	Logra una síntesis precisa del problema a resolver con el desarrollo del sistema.	Logra una síntesis parcial del problema a resolver con el desarrollo del sistema.	La descripción y síntesis del problema usan el mismo texto del instructivo.	Carece de una síntesis del problema abordado por la propuesta de diseño.	
3. Gestión de los requerimientos.	10%	Detalla los requerimientos del sistema producto del análisis del problema y se descomponen los requerimientos dados en el instructivo para derivar requerimientos más específicos siguiendo un formato estándar.	Se descomponen los requerimientos dados en el instructivo para derivar requerimientos más específicos.	Se listan los requerimientos dados en el instructivo	No se consideran los requerimientos como dentro de la propuesta de diseño.	
4. Vista operacional del sistema.	15%	Se presenta y se describe en detalle un concepto de operaciones del sistema, diagrama de casos de uso y secuencia para los segmentos y elementos del sistema.	Se presenta pero no se describe con suficiente nivel de detalle un concepto de operaciones del sistema que considera los segmentos y elementos del sistema.	Se describen los principios de operación del sistema a nivel general.	No se describe, ni presenta un diagrama que describa la operación del sistema propuesto.	
5. Vista funcional del sistema.	10%	Se ilustra y describe completamente la descomposición del sistema considerando las funcionalidades identificadas de acuerdo al análisis de requisitos y el concepto de operaciones propuesto.	Se ilustra y describe parcialmente la descomposición del sistema considerando las funcionalidades identificadas de acuerdo al análisis de requisitos y el concepto de operaciones propuesto.	Se listan algunas funcionalidades del sistema y se describen a nivel general	No se realiza ningún tipo de análisis funcional del sistema	
6. Arquitectura del sistema propuesto.	10%	Se ilustra y describe un diagrama que mapea las funciones e interfaces del sistema en componentes de software y hardware descritas por los requerimientos	Se ilustra y describe un diagrama parcial que mapea las funciones e interfaces del sistema en componentes de software y hardware	Se muestra un diagrama de bloques general que ilustra la interacción de algunos componentes del sistema.	No se presentan, ni describe el mapeo de funciones a componentes de hardware o software.	
7. Análisis de dependencias.	5%	Se ilustra un árbol de dependencias donde se presenten y describan los paquetes de software necesarios y sus relaciones para la implementación de la imagen del sistema operativo con yocto	Se analizan e identifican los paquetes de software requeridos para el desarrollo de la imagen del sistema operativo con yocto	Se mencionan a nivel general las posibles dependencias de software necesarias para la implementación de la imagen del sistema operativo con yocto	No se aborda el tema de dependencias en la propuesta de diseño.	
8. Estrategia de integración de la solución.	15%	Se ilustra y describe una arquitectura integrada de hardware y software para la síntesis de la solución final.	Se describe el proceso integración de los componentes de hardware y software del sistema.	Se menciona vagamente la forma en que los componentes de hardware y software se complementan	No se describe el proceso de integración de componentes de hardware y software para la solución final.	
9. Planeamiento de la ejecución.	5%	Se incluye un diagrama de gantt y una lista de actividades e hitos para el desarrollo del proyecto.	Se describe una lista de productos y entregables a lo largo del desarrollo del proyecto.	Se mencionan vagamente algunos hitos del proyecto pero no se describen.	No se documenta una estrategia de gestión del tiempo para el proyecto.	
10. Conclusiones o aspectos a resaltar de la propuesta presentada.	10%	Se resumen concretamente los aspectos más relevantes de la propuesta de diseño, así como aspectos considerados para su implementación.	Se resumen a nivel general los aspectos más relevantes presentados en la propuesta de diseño.	Solamente se mencionan aspectos que poco se relacionan con los principales hallazgos descritos en la propuesta de diseño.	No se resumen los aspectos más relevantes	