**Assignment Brief (RQF).** Versión en español

Grado: GDAM Curso:\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| **Nombre del Alumno**  **Número de las unidades didácticas y título** | Félix García Miranda |
| Entornos de desarrollo |
| **Año académico**  **Asesor de la unidad didáctica**  **Título de la asignación** | 20/21 |
| Sonia Merayo González |
| Entornos de desarrollo |
| **Fecha de publicación**  **Fecha de presentación** | 09/05/2021 |
| 24/05/2021 |
| **Nombre del Verificador interno**  **Fecha** | Roberto Cruzado Pascual |
|  |

|  |
| --- |
| **Formato de presentación** |
|  |
| **Resultados de aprendizaje** |
| LO1 LO2 LO3 LO4 |
| **Descripción del proyecto y guía** |
| **Contenidos básicos**:  LO1  - Concepto de programa informático.  - Código fuente, código objeto y código ejecutable.  - Tipos de lenguajes de programación.  - Ciclo de vida del programa.  - Instalación de un entorno de desarrollo.  - Funciones de un entorno de desarrollo.  - Uso básico de un entorno de desarrollo.  LO2  - Diagramas de clases: atributos, métodos, visibilidad, instanciación y relaciones.  - Diagramas de casos de uso: actores, sistema y relaciones.  - Diagramas de secuencia: Línea de vida de un objeto, activación y envío de mensajes.  LO3  -Planificación de Pruebas.  - Tipos de pruebas: Funcionales, estructurales, regresión...  - Procedimientos y casos de prueba.  - Pruebas de Código: Cubrimiento, valores límite, clases de equivalencia...  - Pruebas unitarias; herramientas.  LO4  - Control de versiones.  - Herramientas de control de versiones.  - Repositorios.  - Documentación.  - Uso de comentarios.  **Objetivos:**  LO1  - Identificar las fases de desarrollo de una aplicación informática.  - Distinguir entre código fuente, código objeto y ejecutable.  - Clasificar los lenguajes de programación.  - Instalar un entorno de desarrollo.  - Configurar y personalizar un entorno de desarrollo.  - Identificar características comunes y específicas de varios entornos de desarrollo.  - Poder elegir el entorno de desarrollo que mejor se adapte a las necesidades del usuario (sistema, lenguaje, etc.).  LO2  - Identificar los elementos principales de la programación orientada a objetos.  - Utilizar un entorno de desarrollo integrado que permita el diseño de diagramas de clases.  - Ser capaz de definir un diagrama de clases a partir de los requisitos del análisis.  - Poder generar código a partir del diagrama de clases.  - Utilizar ingeniería inversa para generar diagramas de clases a partir de código.  - Conocer la nomenclatura de los diagramas de casos de uso  - Ser capaz de elaborar diagramas de casos de uso.  - Definir diagramas de secuencia para los diferentes casos de uso.  LO3  - Identificar los diferentes tipos de pruebas.  - Poder definir casos de prueba.  - Conocer las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.  - Saber definir puntos de ruptura y seguimiento.  - Poder utilizar las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.  - Efectuar pruebas unitarias de clases y funciones.  - Detectar incidencias.  LO4  - Conocer herramientas de control de versiones  - Saber utilizar una herramienta de control de versiones  - Trabajar con varias ramas dentro de un repositorio.  - Poder utilizar repositorios locales y remotos.  - Conocer las herramientas del entorno de desarrollo para documentar las clases.  - Realizar documentación técnica utilizando comentarios de documentación y etiquetas |

|  |  |  |  |
| --- | --- | --- | --- |
| Guía de calificación | | | |
| **LO1**  **P1.** Elegir un IDE apropiado para un escenario.  **P2.** Redactar un manual de usuario de un IDE destinado a programadores Java.  **M1.** Investigar otros entornos de desarrollo que se puedan usar en el desarrollo de aplicaciones Java.  **D1.** Evaluar si sólo con un IDE concreto se podrían llevar a cabo todas las fases del ciclo de vida del software o, por el contrario, sería necesario software adicional para alguna de ellas.  **LO2**  **P3.** A partir del escenario planteado. Modelar el diagrama de clases apropiado para el modelo de negocio descrito.  **P4.** A partir del escenario planteado. Modelar el diagrama de casos de uso teniendo en cuenta todas las funcionalidades del sistema y los usuarios finales que interactúan con el mismo.  **M2.** A partir del diagrama de casos de uso planteado en el objetivo P4, modelar los diagramas de secuencia que describen el ciclo de ejecución completo de cada caso de uso.  **D2.** Generar un proyecto Java a partir del diagrama de clases modelado en el objetivo P3. Evaluar el resultado obtenido y comprobar que la estructura de clases es la apropiada. En caso contrario, hacer las correcciones necesarias en el diagrama de clases y repetir el proceso.  **LO3**  **P5.** Investigar y explicar los distintos tipos de pruebas de software que existen.  **P6.** Realizar los test unitarios necesarios para una clase Java dada teniendo en cuenta los requisitos del sistema y de los usuarios finales.  **M4.** Documentar el proyecto generado en el objetivo P6 utilizando JavaDoc.  **M5.** Utilizar la herramienta de depuración del IDE sobre los métodos que no pasaron los test del objetivo P6. Realizar las correcciones necesarias y repetir la ejecución de los test.  **D2.** Utilizar un repositorio Git. Se utilizará una rama nueva para cada test realizado.  **LO4**  **P7.** Redactar un documento en el que se incluyan capturas con los resultados de los test realizados en el objetivo P6 e incluir las conclusiones a las que se ha llegado tras evaluar cada uno de los resultados.  **M6.** Redactar un documento en el que se incluyan capturas de la depuración en las que se vea el estado de las variables antes y después de la corrección.  **D3.** Responder de forma razonada a las cuestiones realizadas para obtener conclusiones finales. | Pass | Merit | Distinction |
|  |  |  |