# UNIVERSIDAD MARIANO GALVEZ CENTRO UNIVERSITARIO DE ANTIGUA GUATEMALA

**FACULTAD DE INGENIERÍA EN SISTEMAS PLAN DIARIO**

|  |  |
| --- | --- |
| 2. VARIANTE | **A** |
| **Absoluta:** | 20 Puntos |
| **Relativa:** | 100 Puntos |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Carrera:** | Ingeniería en Sistemas de Información | | | | | | **Código:** | | 1290 |
| **Asignatura:** | Análisis de Sistemas II | | | | | | **Código:** | | 037 |
| **Ciclo:** | Sexto | | | **Fecha:** | | 24/09/2020 | | | |
| **Catedrático:** | Ing. Josué Barillas | | | | | | | | |
| **Semestre:** | 2 | **Sección:** | A | | **Duración del Examen:** | | | 120 min | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Alumno/a: Donal Guillermo García Jauregui** | | | | | | | | | **Carné: 0910-16-8065** | | | |
| 1. ID | | | | | | | | | **Vo.Bo.** | | | |
| 3. EXAMEN | Primer Parcial |  | Segundo Parcial |  | Final | |  | Recuperación | |  | Extraordinario |  |
| 4. INSTRUCCIONES  Conteste o desarrolle cada una de las preguntas o temas planteados en su cuadernillo y anote su respuesta final con lapicero. Caso contrario se invalidará su respuesta. | | | | | | 5. PUNTUACIÓN Y VALORACIÓN | | | | | | |

I SERIE - Resuelva el problema que se le plantea a continuación. Instrucciones generales

|  |  |
| --- | --- |
| I SERIE | 100 puntos |
|  |  |
| Total examen | 100 puntos |

* El examen podrá realizarlo de forma colaborativa en grupos de 4 personas o menos.
* La entrega es individual. Cada estudiante deberá llenar este documento y subirlo a Blackboard.
* Fecha de inicio Lunes 21
* Fecha de finalización Jueves 24
* Hora límite para entregar 23:59. A las 00:00 del día Viernes el sistema ya no permitirá adjuntar ningún documento.
* El examen se deberá entregar por medio de Blackboard únicamente. Herramientas adicionales serán utilizadas únicamente para efectos de calificación pero no tendrán ninguna validez más que para verificar el correcto funcionamiento de la solución.
* Debe dejar constancia de sus scripts, capturas de pantalla y configuraciones en este documento.

Instrucciones generales

* Deberá utilizar GitHub para versionar su código fuente.
* El nombre del repositorio a utilizar será **AS2\_P2**
* El repositorio deberá ser **público**
* El “git log” deberá demostrar todas las operaciones sobre el repositorio incluyendo los commits y los merge.
* La gráfica Insights > Network deberá mostrar las operaciones branch, merge y commits de su proyecto a lo largo del tiempo.
* Deberá utilizar Atlassian Jira para administrar su proyecto
* Planifiquen la duración de cada sprint
* Identifiquen los stories necesarios para resolver el problema y coloquelas en el backlog
* Distribuyan los stories entre los miembros del equipo
* Deberá asignar roles a cada miembro del equipo
* La base de datos a utilizar será de uso común con otros equipos por lo que deberá tener cuidado al manejar la información que se encuentra ahí
* Para acceder a la base de datos utilice las siguientes credenciales:

|  |  |
| --- | --- |
| Driver class name | com.mysql.cj.jdbc.Driver |
| Server | 192.185.4.65 |
| Port | 3306 |
| Database | jbarilla\_as2\_equipo1 jbarilla\_as2\_equipo2 jbarilla\_as2\_equipo3 jbarilla\_as2\_equipo4 jbarilla\_as2\_equipo5 |
| Username | jbarilla\_as2 |
| Password | agEO5ZTI.VC8 |

* Se sugiere utilizar JetBrains Datagrip o MySQL Workbench para gestionar la base de datos
* Se sugiere utilizar JetBrains IntelliJ o Netbeans 12 para su código fuente de backend
* Se sugiere utilizar JetBrains Webstorm o Visual Studio Code -VSC- para su código fuente de frontend Descripción del problema
* Planifique una aplicación de mantenimiento de registros para cualquier tema que implique una estructura en base de datos maestro-detalle. Por ejemplo permisos por rol, animales en un zoológico, médicos en un hospital, etc. No se permite ningun tema relacionado al contexto académico como estudiantes, catedráticos, cursos, etc.
* Las operaciones de mantenimiento deberán incluir las siguientes operaciones:
  + Maestro: crear, leer, modificar, eliminar
  + Detalle: crear, leer, modificar, eliminar
  + Asignar detalle a maestro
  + Desasignar detalle de maestro
  + Ver los detalles del maestro
* Cada tabla deberá tener un mínimo de 10 campos, por ejemplo edad del animal, placa del vehículo, número de licencia, fecha de nacimiento, teléfono, dirección, etc.
* Cada tabla deberá tener al menos un campo que se pueda representar con cada uno de los siguientes controles en el frontend:
  + Text field
  + Combo box
  + Radio button
  + Checkbox
  + Text area
  + Date and time input
  + Number input (spinner)
  + Toggle button (switch)
* Elabore el diagrama de casos de uso en StarUML o cualquier herramienta de diseño.
* Elabore el diagrama de clases de la solución utilizando una arquitectura MVC. El diagrama deberá trabajarlo en una herramienta para UML como StarUML o cualquier otra que permita generar el código desde el diagrama. Esto deberá demostrarlo durante la calificación.
* Para el backend podrá utilizar WebServlets o Spring MVC. No olvide utilizar patrones de diseño como Singleton, DAO, DTO, Factory, Decorator, Null Object, etc. Distribuya sus componentes en paquetes para facilitar el mantenimiento de código.
* Para el backend podrá utilizar Java, C#, PHP, etc. No hay ninguna restricción de lenguaje.
* Para el frontend debe utilizar HMTL, Javascript y CSS. Se sugiere utilizar un framework de desarrollo Javascript tal como:
  + Angular
  + Vue
  + React
  + Stencil
  + Flutter
* El desarrollo debe estar orientado a una aplicación Web, no a Mobile.
* Para la interacción entre el frontend y los servicios en el backend se sugiere utilizar Axios.
* Ponga especial atención en los siguientes aspectos de calidad y usabilidad:
  + No código repetido o duplicado
  + Documentación interna de acuerdo al lenguaje a utilizar (JavaDocs, etc.)
  + Manejo de todos los errores (no try sin catch o finally cuando sea necesario)
  + Manejo de excepciones personalizadas
  + Nombres descriptivos para atributos de clase, variable, métodos, parámetros, clases, etc.
  + Cerrar todos los recursos al terminar de utilizarlos (archivos, conexiones a bases de datos, etc.)
  + Menor cantidad posible de clics para el usuario
  + Evitar los scrolls en la página para listas muy largas
  + Tiempos de espera (si el usuario debe esperar mostrar una imagen “cargando…”, gif, barra de progreso, bloquear pantalla, etc.)
* El backend y el frontend estarán publicados en dos servidores diferentes. Es decir, que será una aplicación distribuida. Por ejemplo el frontend en un Apache Web Server o IIS y el backend en un Glassfish o Weblogic.

Complete

1. Integrantes del equipo

|  |  |
| --- | --- |
| No. Carné | Nombre Completo |
| 0910-02-13118 | Danilo Estuardo Itzep Luna |
| 0910-16-8065 | Donal Guillermo García Jauregui |
| 0910-04-12850 | Samuel Felipe Solís |
|  |  |

1. Rol de cada miembro del equipo

|  |  |
| --- | --- |
| Nombre | Rol (project manager, backend dev, frontend dev, database dev, etc.) |
| Donal García | Project Manager |
| Samuel Felipe | Database dev |
| Danilo Estuardo Itzep Luna | Backend Dev, Frontend Dev |
|  |  |

1. Diagrama de casos de uso

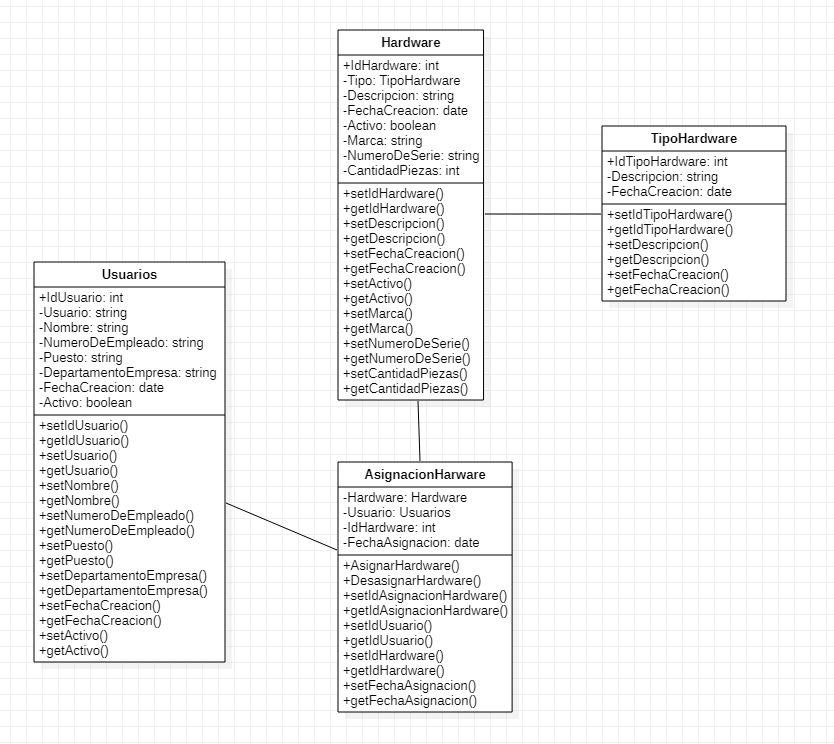
Imagen que contiene texto, mapa

Descripción generada automáticamenteImagen que contiene texto, mapa

Descripción generada automáticamenteImagen que contiene texto, mapa

Descripción generada automáticamente

1. Diagrama de clases



1. Describa los aspectos de usabilidad utilizados

|  |  |
| --- | --- |
| Concepto | Descripción |
| Intuitivo | El maestro guía de forma práctica al usuario, utilizando botones con gráficos alusivos a su función. |
| Visibilidad del estado del registro | Detalle de mensajes de los aciertos y errores al grabar un registro. |
| Consistencia | El diseño del maestro en su navegación, estándares de botones, orden de tabulación. Todo esto se respeta en las pantallas. |
| Rapidez | Todas las pantallas se muestran con la mayor rapidez posible, asimismo se agrega mensaje de loading mientras carga la data. |
| Simple | El diseño de la página web es simple y de fácil acceso. |
| Compatibilidad | La página web, funciona en los 3 principales navegadores, Google Chrome, Internet Explorer, Mozilla Firefox. |

1. Repositorio

**https://github.com/Djauregui77/AS2\_P2**

Calificación

|  |  |  |
| --- | --- | --- |
| Descripción | Puntos posibles | Puntos obtenidos |

|  |  |  |
| --- | --- | --- |
| Diagrama de casos de uso | 1 |  |
| Diagrama de clases y código fuente generado 100% funcional | 2 |  |
| Aspecto visual de la aplicación (interfaz gráfica de usuario) | 1 |  |
| Usabilidad | 2 |  |
| Documentación interna de código fuente | 3 |  |
| Manejo de errores y retroalimentación al usuario | 3 |  |
| Operaciones básicas del maestro detalle (crear, ver, modificar, eliminar) | 4 |  |
| Operaciones avanzadas del maestro detalle (agregar/quitar detalle) | 2 |  |
| Tolerancia a todo tipo de errores | 2 |  |