



Guion de Prácticas

Programación II
1º Grado en Ingeniería de Sistemas de Telecomunicación
Universidad CEU San Pablo
Sergio Saugar García

Abril 2022



Jurassic Park Simulator, build your own park!

ENTREGA OBLIGATORIA: SI

FECHA LÍMITE DE ENTREGA: 3 de Junio de 2022 (Primera Entrega: 24 de Abril)

PESO: 100 % de la nota de prácticas (Lea los pesos asignados a cada entrega parcial)

Esta práctica es de **entrega obligatoria**. Tenga en cuenta que:

- La sección 4 recoge instrucciones sobre qué elementos deben ser entregados y, también, instrucciones sobre el procedimiento a seguir.
- La fecha límite para la entrega de la práctica está establecida en este enunciado. No se aceptarán entregas a partir de dicha fecha.

Tenga en cuenta que el enunciado de esta práctica establece **un sistema de entregas parciales** (descrito en la **sección 4**).

- Cada entrega parcial contribuye a la nota final de la práctica según el peso indicado.
- **Para que las entregas parciales sean evaluadas** deberán enviarse dentro del plazo establecido (en caso contrario, **perderá la nota correspondiente a la entrega no enviada/ enviada fuera de plazo**).

Por último, es **obligatorio** el uso de un repositorio *GitHub* (en el que el profesor aparezca como colaborador), **para que la práctica sea evaluable**. En la sección 3 se establece **cómo se espera que el alumno utilice el repositorio para que la práctica sea tomada en consideración**.

1. Introducción

Esta práctica tiene como objetivo principal la consolidación de los conocimientos relacionados con el paradigma de Programación Orientada a Objetos, algoritmos de búsqueda y ordenación y con el manejo de estructuras de datos complejas. En ese sentido, se desarrollará como aplicación un hipotético juego acerca de una conocida saga de películas, utilizando el lenguaje de programación *Java*.



Para la realización de esta práctica, es necesario que:

- Si no lo ha hecho todavía, descargue y configure el IDE *Netbeans*¹.
- **Cree un repositorio privado en *GitHub*** para gestionar **exclusivamente** la práctica (encontrará más información sobre cómo hacerlo y qué se espera de cómo usarlo en la sección 3.1).
- Una vez creado el repositorio, genere un proyecto de *Netbeans* única y exclusivamente para la práctica. **El nombre del proyecto** deberá comenzar **OBLIGATORIAMENTE** de la siguiente manera: la **inicial de su nombre**, su **primer apellido completo** y la **inicial de su segundo apellido** (por ejemplo: *ssaugarg*). Después agregue el texto que desee.
- Realice commits cada vez que introduzca un cambio en el código y súbalo a *GitHub* (la política de commits que se valorará está establecida en el apartado 3.1).
- Desde el primer momento, y antes de empezar a codificar el proyecto, gestione los permisos del repositorio *GitHub* para agregar al profesor de la asignatura como “colaborador” (el usuario del profesor es: *ssaugar*).
- Tenga en cuenta que **el uso de un repositorio privado en *GitHub* para gestionar esta práctica es obligatorio** y servirá:
 - En primer lugar, para **poder realizar sesiones de tutorías *online* con el profesor** de la asignatura de una manera más eficaz.
 - En segundo lugar, **se tendrá en cuenta para la evaluación de la práctica** (lea atentamente la sección 3).

A continuación, en la **sección 2** se establece el **contexto y los requisitos** que debe cumplir la aplicación. La **sección 3** especifica cuestiones propias de la **evaluación de la práctica** y **el uso que se espera que el alumno haga del repositorio *GitHub***. Posteriormente, la **sección 4** especifica: **las fechas de las entregas parciales, la nota asignada a cada una de ellas, los elementos que deben enviarse y cómo debe realizarse el envío al profesor**. Finalmente, la **sección 5** establece los **mecanismos por los que se resolverán las dudas** correspondientes a esta práctica.

2. Enunciado

Como parte de su formación en el Grado en Ingeniería de Sistemas de Telecomunicación, va a comenzar a realizar prácticas en la empresa *Crichton Games* que se dedica a programar juegos para diversas plataformas online.

Recientemente, *Crichton Games* ha comenzado el desarrollo de un juego acerca de una conocida saga de *Hollywood*, denominado:

Jurassic Park Simulator, build your own park!

Esperan que el juego sea todo un revulsivo en el mercado actual y les permita conseguir financiación suficiente para ampliar el negocio y, quién sabe, quizá vender la empresa y poder retirarse!

Es la primera vez que van a desarrollar un juego para la plataforma *Java*, ya que, hasta el momento, sólo habían desarrollado juegos para móviles. Precisamente por eso, al ver sus buenas

¹*Netbeans* . <https://netbeans.apache.org/download/index.html>



notas en *Programación II*, el jefe de proyecto decidió seleccionarle para realizar las prácticas. Pensó que no sólo podría ayudarles en las labores de programación, sino también en las de diseño del futuro software. Y, como para usted suponía un reto personal poner en práctica aquellos conceptos adquiridos en la asignatura, sin pensárselo dos veces, aceptó la propuesta.

Tras varias tormentas de ideas con los creativos de la empresa, el jefe de proyecto le hace entrega de un dossier con una explicación informal de los requisitos funcionales que debería reunir el juego. También le informa de la fecha en la que usted debería tener listo un prototipo totalmente funcional de la aplicación.

Parece que se ha embarcado en una buena aventura...

2.1. Requisitos funcionales

Los creativos de *Crichton Games* están pensando en un juego que permitirá ponerse en la piel de *John Hammond*² un famoso filántropo multimillonario interesado en la investigación genética.

El interés de *John Hammond* por los dinosaurios, se remonta a 1975, cuando fundó *International Genetic Technologies Inc.*³, más conocida como *InGen*, una *start-up* biotecnológica que centraría sus recursos en investigar cómo recrear especies extinguidas. Así, a principios de los 80, *Hammond* fantaseaba con la idea de clonar dinosaurios a partir de *ADN* preservado en ámbar fosilizado. Para ello, creó un equipo de investigadores dirigido por el *Dr. Henry Wu*⁴, ingeniero jefe de genética (*Chief Genetic Engineer*) en *InGen*. El objetivo inicial fue llevar a cabo estas investigaciones dentro de la “*Hammond Foundation*” de una manera altruista, pero pronto se descubriría el potencial económico del descubrimiento. En 1985, *John Hammond* había abierto *InGen* a inversores externos, fortaleciendo así la empresa y creando la primera y única empresa mundial, generadora de conocimiento puntero y líder en el campo de la ciencia y la tecnología genética y biológica. Y, así, finalmente en 1986, lograron clonar el primer animal prehistórico.

El cuartel general de *International Genetic Technologies Inc.* estaba localizado en *Palo Alto (California)*, y allí se llevó a cabo toda la investigación inicial. Sin embargo, más tarde, todas las instalaciones de investigación se trasladaron a la isla *Isla Sorna*, a 333 kilómetros de *Costa Rica*. Allí criaban a los dinosaurios que recreaban.

Tras el éxito clonando diversas especies de dinosaurios, *Hammond* se planteó la construcción de un sitio donde el público pudiera observar estos bellos e imponentes animales. Así, en algún momento de la década de los ochenta, se comenzó la construcción de un anfiteatro en la ciudad de *San Diego*. Sin embargo, en 1988 se abandonó la idea del anfiteatro y se comenzó la construcción de un parque de atracciones jurásico en la isla *Isla Nublar*, a 194 kilómetros de *Costa Rica*.

Sin embargo, en 1993, cuando el parque estaba cerca de su inauguración, la intromisión de una empresa rival (seguramente *BioSyn*⁵, causó la caída de todos los sistemas informáticos de la isla y los dinosaurios se escaparon de sus localizaciones. Esto supuso un duro golpe para *International Genetic Technologies Inc.*, con la pérdida de confianza de sus inversores, *Jurassic Park* no llegó a abrir sus puertas.

En este contexto, los creativos de *Crichton Games* han pensado que *Jurassic Park Simulator, build your own park!*, podría ser el típico juego de construcción de parques de atracciones (¡¡¡qué creativos más originales!!!). Lo único es que, en vez de existir un tiovivo con caballitos de cartón piedra, nos podremos subir a lomos de un gracioso *Velociraptor*⁶. O, en vez de visitar la instalación

² *John Hammond*. https://jurassicpark.fandom.com/wiki/John_Hammond

³ *International Genetic Technologies Inc.* (*InGen*). <https://jurassicpark.fandom.com/wiki/InGen>

⁴ *Dr. Henry Wu*. https://jurassicpark.fandom.com/wiki/Henry_Wu

⁵ *Biosyn*. <https://jurassicpark.fandom.com/wiki/Biosyn>

⁶ *Velociraptor*. <https://jurassicpark.fandom.com/wiki/Velociraptor>



de los delfines, nos podremos bañar alegremente con unos *Platecarpus*⁷.

Para ello, han definido el esquema general del juego que se describe a continuación.

En cada *partida*, un *jugador* selecciona un *escenario* que establece el nivel de dificultad de la misma. Las partidas tienen un *identificador* único, de forma que el juego pueda tener almacenadas distintas partidas a punto de comenzar o ya finalizadas (a modo de histórico).

Los *escenarios* (sección 2.1.1) se corresponden con localizaciones famosas del universo *Jurassic Park*. Un *escenario* simplemente es un elemento del juego que establece para el jugador un número de *monedas* iniciales con las que deberá criar a sus dinosaurios (sección 2.1.4), y también establecerá las características de la isla en la que construirá su parque temático (2.1.2). Los escenarios también establecen el número de *visitantes* y su crecimiento o decrecimiento a lo largo de la vida del parque.

Así, el objetivo del *jugador* es lograr lo que no consiguió *John Hammond* : abrir y mantener vivo *Jurassic Park* durante el máximo tiempo posible. Para ello, deberá criar sus dinosaurios en la *Isla Sorna* . Una vez estén maduros, deberá construir instalaciones adecuadas en *Jurassic Park* para albergarlos y, posteriormente, los trasladará. Allí, el público asistirá a un maravilloso espectáculo: ver de cerca los grandes animales prehistóricos. Como en todos los parques temáticos, el éxito del parque depende de la experiencia que tenga el visitante. A mejor experiencia más gasto y, a más gasto, más dinero para invertir en criar dinosaurios y más dinosaurios para exponer al público, siguiendo una espiral de éxito. ¿Qué puede salir mal?

De esta manera, los jugadores tienen las siguientes características:

- Cada jugador tiene un *alias* dentro del juego que es un identificador único.
- Cada jugador posee dos islas: una isla *de crianza* y otra isla *de exhibición*. En ellas clona los dinosaurios y los expone en su *Jurassic Park* (respectivamente).
- Cada jugador posee un número de *monedas* que va ganando a lo largo del juego y que utiliza para configurar y mejorar su parque y sus dinosaurios.
- Además, cada jugador cuenta con un *centro de mando* donde puede ver qué está pasando con en sus islas y el histórico de la partida. Esto le permitirá tomar decisiones.

Por su parte, existen dos tipos de islas: isla *de crianza* e isla *de exhibición*.

- La isla *de crianza* (*Isla Sorna*) se utiliza como granja de producción de dinosaurios. En ella, el usuario puede crear nuevos dinosaurios y alimentarlos hasta que sean adultos. En este tipo de isla, el jugador crea instalaciones donde los animales son clonados y mantenidos hasta su edad adulta. Una vez que son adultos, el usuario los puede traspasar a la isla *de exhibición* (por ejemplo, *Isla Nublar*) donde se muestran al público.
- La isla *de exhibición* se utiliza como parque temático donde se exponen los dinosaurios que el jugador ha criado. En ella, el jugador crea instalaciones donde los animales son retenidos y alimentados.

Las *islas* poseen un cierto número de *hectáreas* que el jugador podrá utilizar, *instalaciones* *de exhibición* (o *crianza*), una cantidad determinada de *comida* y cierta población de *dinosaurios* (sección 2.1.4).

⁷ *Platecarpus*. <https://jurassicpark.fandom.com/wiki/Platecarpus>



Durante el transcurso de la partida, los jugadores pueden controlar el tránsito de dinosaurios entre sus islas. El objetivo final de cada jugador es coleccionar el mayor número posible de ejemplares en su *Jurassic Park*, manteniéndolos saludables y lustrosos para que los visitantes los seleccionen como favoritos (y así, ganar monedas).

La partida no podrá comenzar hasta que el jugador no haya elegido un escenario y configurado su isla de crianza.

Después de esta introducción, en las siguientes secciones se describe, más detalladamente, cada uno de los elementos del juego.

2.1.1. Escenarios del juego

Las *partidas* siempre se desarrollan en el ámbito de un *escenario* del juego. Cada *escenario* restringe las condiciones iniciales del jugador, estableciendo el número de *monedas* disponible para comenzar la partida y la isla de exhibición que el jugador utilizará (la isla de crianza siempre será *Isla Sorna*).

La configuración básica del juego incluirá los siguientes escenarios:

- *Jurassic Park*⁸. Un escenario real, basado en las películas que has vivido. Este escenario establece una cantidad de *100000 monedas* iniciales para el jugador y la *Isla Nublar* como isla de exhibición. Esta isla se encuentra a muchas millas de la población, sólo gente de gran poder adquisitivo se puede permitir visitar un parque tan remoto. ¿Quién no querría jugar en este escenario siendo el sitio donde comenzó toda la historia?
- *Jurassic Park: San Diego*⁹. Un limitado anfiteatro para poder instalar unas cuantas muestras del gran período prehistórico. De pequeñas dimensiones pero en medio de una gran ciudad. Este escenario establece una cantidad de *50000 monedas* iniciales para el jugador y los terrenos de *San Diego* como isla de exhibición (no es una isla, pero ¿a quién le importa?). Muchísimos norteamericanos se animarán a visitar tu parque, ¿podrás ir variando los animales que exhibes para proporcionar una inolvidable experiencia?
- *Jurassic World: Evolution*¹⁰. Este escenario te propone crear un nuevo parque temático en la *Isla Matanceros*. Olvídate de las ruinas que quedaron en la *Isla Nublar* después del Huracán Clarissa. Construye un nuevo parque en una isla virgen. Para ello dispondrás de *150000 monedas*. Esta isla se encuentra en el mismo archipiélago que la *Isla Sorna* (Archipiélago de Muertes¹¹) pero tiene mejor conexión marítima con *Costa Rica*, lo que te proporcionará un mayor flujo de visitantes.

2.1.2. Islas

Las islas determinan la cantidad de espacio y los recursos necesarios para realizar las actividades de crianza y exhibición de los dinosaurios. Así, las islas tienen un número de *hectáreas* en el que pueden desarrollar la actividad principal. Dentro de ese terreno contienen un conjunto de *instalaciones* que clonan o sirven para exhibir a los dinosaurios. Por lo que, al final, las islas poseen una *población* de dinosaurios conviviendo en ellas. Para la supervivencia de los animales, las islas almacenan una cantidad de *comida* que sirve para alimentar a los animales.

⁸[https://jurassicpark.fandom.com/wiki/Jurassic_Park_\(film\)](https://jurassicpark.fandom.com/wiki/Jurassic_Park_(film))

⁹https://jurassicpark.fandom.com/wiki/Jurassic_Park:_San_Diego

¹⁰https://jurassicpark.fandom.com/wiki/Isla_Matanceros

¹¹https://jurassicpark.fandom.com/wiki/Muertes_Archipelago



En el caso de las islas de exhibición, poseen dos atributos adicionales: número de *visitantes* mensuales y poder *adquisitivo* de los visitantes (que comprende: alto, medio, bajo). Es cierto que estos números se irán ajustando según vaya progresando la partida, dependiendo de factores como la variedad y calidad de los animales que se exhiban.

La configuración básica del juego proporcionará, al menos, las siguientes islas:

- *Isla Sorna* :
 - *Actividad*: CRIANZA
 - *Alimento*: 20000 unidades inicialmente
 - *Hectáreas*: 5000
- *Isla Nublar* :
 - *Actividad*: EXHIBICIÓN
 - *Alimento*: 10000 unidades inicialmente
 - *Hectáreas*: 7700
 - *Visitantes*: 1000
 - *NivelAdquisitivo*: ALTO
- *San Diego* :
 - *Actividad*: EXHIBICIÓN
 - *Alimento*: 5000 unidades inicialmente
 - *Hectáreas*: 100
 - *Visitantes*: 2000
 - *NivelAdquisitivo*: BAJO
- *Isla Matanceros* :
 - *Actividad*: EXHIBICIÓN
 - *Alimento*: 25000 unidades inicialmente
 - *Hectáreas*: 10000
 - *Visitantes*: 3000
 - *NivelAdquisitivo*: MEDIO

2.1.3. Instalaciones

En las islas se pueden construir instalaciones que permitan conseguir su objetivo. De esta manera, las instalaciones se dividen en *instalaciones de cría* e *instalaciones de exhibición*. Cada instalación tiene un coste de construcción, ocupa ciertas *hectáreas* y es adecuada para dinosaurios que habitan un *medio* concreto (terrestre, volador, acuático) y una *alimentación* concreta (carnívoro, herbívoro). Además, sólo pueden mantener a un *número de ejemplares* concreto. La *alimentación* de los dinosaurios también es proporcionada por la instalación. El tipo de instalación dependerá de las características de sus habitantes (instalación de madera, instalación con verja electrificada, instalación con visor de vidrio).

La calidad de vida de los dinosaurios depende directamente de las condiciones en que se encuentre la instalación en la que habitan.

La configuración básica del juego proporcionará, al menos, las siguientes islas:



- Instalaciones de cría:
 - Instalación CTH05:
 - *Coste de construcción*: 1000 monedas
 - *Tipo de medio*: TERRESTRE
 - *Tipo de alimentación*: HERBIVORO
 - *Unidades de alimentación*: 200 unidades inicialmente
 - *Número de ejemplares*: 5
 - *Hectáreas*: 10
 - *Tipo de recinto*: MADERA
 - Instalación CTC03:
 - *Coste de construcción*: 2500 monedas
 - *Tipo de medio*: TERRESTRE
 - *Tipo de alimentación*: CARNÍVORO
 - *Unidades de alimentación*: 300 unidades inicialmente
 - *Número de ejemplares*: 3
 - *Hectáreas*: 50
 - *Tipo de recinto*: ELECTRIFICADO
 - Instalación CVC02:
 - *Coste de construcción*: 5000 monedas
 - *Tipo de medio*: VOLADOR
 - *Tipo de alimentación*: CARNÍVORO
 - *Unidades de alimentación*: 500 unidades inicialmente
 - *Número de ejemplares*: 2
 - *Hectáreas*: 100
 - *Tipo de recinto*: ELECTRIFICADO
 - Instalación CAC4:
 - *Coste de construcción*: 3500 monedas
 - *Tipo de medio*: ACUÁTICO
 - *Tipo de alimentación*: CARNÍVORO
 - *Unidades de alimentación*: 100 unidades inicialmente
 - *Número de ejemplares*: 4
 - *Hectáreas*: 25
 - *Tipo de recinto*: VIDRIO
- Instalaciones de exhibición:
 - Instalación ETH02:
 - ◇ *Coste de construcción*: 21000 monedas
 - ◇ *Tipo de medio*: TERRESTRE
 - ◇ *Tipo de alimentación*: HERBIVORO
 - ◇ *Unidades de alimentación*: 0 unidades inicialmente
 - ◇ *Número de ejemplares*: 2
 - ◇ *Hectáreas*: 200
 - ◇ *Tipo de recinto*: MADERA
 - Instalación ETC02:



- ◊ *Coste de construcción*: 12500 monedas
- ◊ *Tipo de medio*: TERRESTRE
- ◊ *Tipo de alimentación*: CARNÍVORO
- ◊ *Unidades de alimentación*: 0 unidades inicialmente
- ◊ *Número de ejemplares*: 3
- ◊ *Hectáreas*: 300
- ◊ *Tipo de recinto*: ELECTRIFICADO
- Instalación EVC02:
 - ◊ *Coste de construcción*: 15000 monedas
 - ◊ *Tipo de medio*: VOLADOR
 - ◊ *Tipo de alimentación*: CARNÍVORO
 - ◊ *Unidades de alimentación*: 0 unidades inicialmente
 - ◊ *Número de ejemplares*: 2
 - ◊ *Hectáreas*: 200
 - ◊ *Tipo de recinto*: ELECTRIFICADO
- Instalación EAC03:
 - ◊ *Coste de construcción*: 25000 monedas
 - ◊ *Tipo de medio*: ACUÁTICO
 - ◊ *Tipo de alimentación*: CARNÍVORO
 - ◊ *Unidades de alimentación*: 0 unidades inicialmente
 - ◊ *Número de ejemplares*: 3
 - ◊ *Hectáreas*: 400
 - ◊ *Tipo de recinto*: VIDRIO

2.1.4. Dinosaurios

En cada una de las islas que controla el jugador, dentro de las instalaciones determinadas para ello, habitan las poblaciones de dinosaurios.

Cada *Dinosaurio* tiene como características principales:

- Un *identificador* único que le proporciona *InGen* en el momento de su nacimiento.
- Un *medio* en el que habita: *TERRESTRE*, *VOLADOR*, *ACUATICO*.
- Tipo de *alimentación*: *CARNIVORO*, *HERBIVORO*
- Su *edad* actual (número de meses desde que nació).
- La edad a la que se hace *adulto*.
- Un porcentaje que determina su nivel de *salud* y que, al nacer, es del 100 %.
- El número de visitantes que le han elegido como *favorito*.

Dinosaurios disponibles y sus características En la versión inicial del juego, los dinosaurios¹² que se definirán serán los siguientes:

- **Nombre: Tyrannosaurus rex**¹³
 - *Medio*: Terrestre

¹²Más dinosaurios en: <https://jurassicpark.fandom.com/wiki/Category:Dinosaurs>

¹³Tyrannosaurus rex. https://jurassicpark.fandom.com/wiki/Tyrannosaurus_rex



- *Alimentación*: Carnívoro
- *Edad adulto*: 24
- *Nombre*: **Velociraptor**¹⁴
 - *Medio*: Terrestre
 - *Alimentación*: Carnívoro
 - *Edad adulto*: 38
- *Nombre*: **Brachiosaurus**¹⁵
 - *Medio*: Terrestre
 - *Alimentación*: Herbívoro
 - *Edad adulto*: 12
- *Nombre*: **Pteranodon**¹⁶
 - *Medio*: Volador
 - *Alimentación*: Carnívoro
 - *Edad adulto*: 16
- *Nombre*: **Mosasaurus**¹⁷
 - *Medio*: Acuático
 - *Alimentación*: Carnívoro
 - *Edad adulto*: 42

2.1.5. Simulación de la partida

Será definido en la segunda entrega.

2.1.6. Centro de mandos

Será definido en la tercera entrega.

2.1.7. Gestión de errores y excepciones

Además del cumplimiento de la funcionalidad establecida, se requiere que el software incluya todos los chequeos (y tratamiento de excepciones) necesarios para asegurar que los datos procesados en la aplicación sean consistentes.

2.1.8. Ampliaciones

Será definido en la tercera entrega.

2.2. Operaciones de entrada/salida de datos

Será definido en la tercera entrega.

¹⁴Velociraptor. <https://jurassicpark.fandom.com/wiki/Velociraptor>

¹⁵Brachiosaurus. <https://jurassicpark.fandom.com/wiki/Brachiosaurus>

¹⁶Pteranodon. <https://jurassicpark.fandom.com/wiki/Pteranodon>

¹⁷Mosasaurus. <https://jurassicpark.fandom.com/wiki/Mosasaurus>



2.3. Interfaz textual/gráfica

Será definido en la segunda entrega.

2.4. Consejos

- Varios puntos del enunciado están abiertos, tome sus decisiones de diseño y coméntelas con el profesor para que las valide.
- Organice y estructure el código usando herencia, interfaces, polimorfismo...
- Establezca la visibilidad de atributos y métodos para asegurar la encapsulación de los datos.
- Reutilice todo el código que pueda (constructores del padre, código de métodos heredados, etcétera).
- Estructure su código en pequeños métodos que realicen funciones concretas.
- Organice el código de la práctica en varios paquetes: las clases que representan datos de la aplicación en el paquete *modelo*, la clase almacén de datos en el paquete *data*, las clases que trabajan con entrada/salida en el paquete *io*, las clases de la interfaz de usuario en el paquete *gui*, etcétera.

3. Evaluación

Para superar la práctica, además del software entregado, el alumno tendrá que realizar una defensa (respondiendo a preguntas formuladas por el profesor sobre funcionamiento, cuestiones de diseño, partes del código, etcétera). Si el alumno **no superase la defensa de la práctica, la nota final de la práctica será de cero puntos**.

Además de la defensa, la práctica se evaluará en relación a:

- **Organización y estructuración del código** (utilización de conceptos y patrones de programación orientada a objetos: herencia, polimorfismo, encapsulación, reutilización, utilización correcta de estructuras de control, ...).
- Utilización de **estructuras de datos adecuadas** y utilización de **técnicas de ordenación y búsqueda**, ... El uso de todas ellas deberá ser analizado y justificado en la memoria descriptiva.
- **Uso de Javadoc** para documentar las clases y métodos.
- Funcionamiento ajustado a los requisitos establecidos (**incluyendo, además de chequeos de datos, integridad de la información, gestión de excepciones...**).
- Si implementa una interfaz textual, la nota de la práctica será valorada sobre 10 puntos.
 - Si implementa una **interfaz gráfica** utilizando *Java Swing*, podrá conseguir **hasta dos puntos extra**.
- Si implementa **alguna de las ampliaciones** recogidas en el apartado 2.1.8 o alguna **ampliación acordada con el profesor**, podrá conseguir hasta dos puntos extra.
- **Claridad del código y adecuación a las normas de estilo** (correcto nombrado de clases, métodos y variables, comentarios internos, indentación del código, ...).

Se advierte al alumno de que, como parte de la evaluación, se utilizará una **herramienta antiplagio** de código. En caso de copia total o parcial del código de la práctica, se aplicarán las medidas correspondientes.



3.1. Utilización del repositorio *GitHub*

Para que la práctica sea evaluable es **obligatoria** la utilización de un repositorio *GitHub*. Allí, el alumno **registrará toda la actividad de desarrollo de la práctica**.

De esta manera, se espera que el alumno vaya realizando *commits* de su trabajo cada vez que realice algún trabajo significativo sobre el software (añadir una nueva clase, implementación de un nuevo método o funcionalidad, modificación de los atributos de una clase,...). Es decir, **el alumno debe realizar *commits* de manera regular** y con un **propósito preciso** (el alumno debe evitar hacer *commits* por el mero hecho de incrementar la cantidad de los mismos; debe hacerlos con el objetivo de registrar un nuevo incremento válido en el software). Se recomienda al alumno que se asegure de que el proyecto compila antes de hacer un *commit*.

Es responsabilidad del alumno realizar los *commits* de manera adecuada, de forma que se pueda evaluar que su trabajo se ha realizado de manera incremental.

Se advierte que las siguientes situaciones: hacer un único (o unos pocos) *commit*, hacer *commits* sin un propósito razonable (sin incremento de software asociado) o realizar una gestión de los *commits* **que no permitan comprobar que el alumno ha trabajado constantemente y de manera incremental** en el proyecto, **pueden derivar en un suspenso de la práctica**.

Por lo tanto, **antes de comenzar a codificar la práctica:**

- Genere un **repositorio privado** en *GitHub*, clónelo en su disco duro y cree un proyecto *Netbeans* en dicha carpeta.
- **El nombre del proyecto** de *Netbeans* deberá ajustarse a lo establecido al inicio de este enunciado.
- Haga un primer *commit* añadiendo el proyecto y súbalo al repositorio remoto.
- Posteriormente, los *commits*, *pull* y *push* podrá realizarlo directamente desde *Netbeans* directamente.
- No se olvide de agregar al profesor como colaborador de su repositorio (el usuario del profesor en *GitHub* es *ssaugar*)

Si tiene dudas acerca de cómo crear o manejar un repositorio de *GitHub*, a principio de curso, se colgó información de su uso en el *Campus Virtual* de la asignatura. Si aun así sigue teniendo dudas, contacte lo antes posible con el profesor de la asignatura.

4. Calendario de entregas

Para lograr la **realización incremental y progresiva de la práctica**, se establece el siguiente calendario de entregas. En cada apartado se indica: qué porcentaje de la nota será evaluado en dicha entrega, qué hitos deben entregarse, cómo se debe realizar la entrega y la obligatoriedad de la misma. Si necesita cualquier tipo de aclaración, consulte con el profesor.

Para aceptar una entrega es **condición necesaria** que el código entregado **compile correctamente**. Si el código presentase errores de compilación, la entrega sería rechazada, no evaluada y calificada con cero puntos.



4.1. Primera entrega - 24 de Abril de 2022

- Entrega **no obligatoria**.
- La evaluación de esta entrega tendrá un **peso del 5 %** de la nota de prácticas.
- **Hito a entregar:** Paquete *modelo*. Es decir, todas las clases que representen datos de la aplicación. Incluirá la estructura de todas las clases (con atributos, constructores y otros métodos), relaciones de herencia ya establecidas, enumerados si los hubiese...

Como resultado de esta entrega intermedia se deberá enviar un fichero *zip* con la carpeta completa que contenga el proyecto software *Netbeans* correspondiente a la práctica (asegúrese de que incluye todo el código fuente).

Además, en su repositorio de *GitHub* deberá realizar un *commit* denominado "*Primera Entrega*." indicando así, el punto temporal exacto de la entrega.

El fichero se enviará a través de la actividad que será creada a tal efecto en el *Campus Virtual* de la asignatura. Recuerde que la **entrega de la práctica deberá realizarse dentro del período de tiempo indicado**, en caso contrario, se evaluará con cero puntos.

4.2. Segunda entrega - 13 de Mayo de 2022

- Entrega **no obligatoria**.
- La evaluación de esta entrega tendrá un **peso del 5 %** de la nota de prácticas.
- **Hito a entregar:** Paquetes *data* y *gui*. Es decir, la clase que representa el almacenamiento de datos de la aplicación y la interfaz de usuario (así como la lógica asociada al juego). En este punto, el usuario ya debería poder jugar con la aplicación.

Como resultado de esta entrega intermedia se deberá enviar un fichero *zip* con la carpeta completa que contenga el proyecto software *Netbeans* correspondiente a la práctica (asegúrese de que incluye todo el código fuente).

Además, en su repositorio de *GitHub* deberá realizar un *commit* denominado "*Segunda Entrega*." indicando así, el punto temporal exacto de la entrega.

El fichero se enviará a través de la actividad que será creada a tal efecto en el *Campus Virtual* de la asignatura. Recuerde que la **entrega de la práctica deberá realizarse dentro del período de tiempo indicado**, en caso contrario, se evaluará con cero puntos.

4.3. Entrega final - 3 de Junio de 2022

- Entrega **obligatoria**.
- La evaluación de esta entrega tendrá un **peso del 90 %** de la nota de prácticas.
- **Hito a entregar:** Implementación íntegra de los requisitos establecidos en el enunciado de la práctica.
- Se entregará también una **memoria descriptiva** del trabajo realizado.

Como resultado de esta entrega final se deberá enviar un fichero *zip* con la carpeta completa que contenga el proyecto software *Netbeans* correspondiente a la práctica (asegúrese de que incluye todo el código fuente).

Además, en su repositorio de *GitHub* deberá realizar un *commit* denominado "*Entrega Final*." indicando así, el punto temporal exacto de la entrega.

El envío de la memoria y el fichero se realizará siguiendo estas instrucciones:



- La memoria incluirá una portada con el nombre del alumno y se ajustará a la siguiente estructura:
 - Análisis y descripción de la aplicación:
 - ◊ Este análisis y descripción dará respuesta a las siguientes preguntas:
 - ◊ Cómo se han organizado y estructurado las clases y cuál es la responsabilidad de cada una.
 - ◊ Qué decisiones de diseño se han tomado.
 - ◊ Qué comprobaciones de integridad (y excepciones) se han implementado.
 - ◊ Qué estructuras de datos ha utilizado y por qué lo ha hecho.
 - ◊ Qué técnicas de ordenación y búsqueda ha utilizado y por qué lo ha hecho.
 - ◊ **Diagramas de clases UML**¹⁸
 - Listado de fallos conocidos y **funcionalidades definidas en el enunciado que no se han implementado** en el código entregado.
 - Listado de todo el código fuente de la aplicación organizado por paquetes y clases.
 - Conclusiones (que incluirán, **obligatoriamente**, una valoración del tiempo dedicado a la práctica).
- La memoria y el fichero se enviarán a través de la actividad que será creada a tal efecto en el *Campus Virtual* de la asignatura.
- Recuerde que la **entrega de la práctica deberá realizarse dentro del período de tiempo indicado**, en caso contrario, se evaluará con cero puntos.

5. Tutorías

Se recomienda que el alumno asista a reuniones periódicas con el profesor para asegurarse de que está estructurando y organizando el código de una manera correcta. Las consultas y dudas que surjan en el desarrollo de esta práctica, se resolverán durante las horas de tutorías. Para ello, es obligatorio concertar una cita con el profesor (a través de correo electrónico).

6. Referencias

Los nombres, personajes, sitios e imágenes que aparecen en este enunciado están registrados por sus respectivos dueños. La información que ha servido para confeccionar este enunciado, ha sido extraída de los siguientes sitios:

1. *Frontier - Jurassic World Evolution (logo)*
<https://www.jurassicworldevolution.com/es-ES>
2. *Jurassic Park Fandom Wiki*
https://jurassicpark.fandom.com/wiki/Jurassic_Park_Wiki

¹⁸Para la realización de los diagramas se recomienda el uso del programa *Magicdraw* <https://www.nomagic.com/products/magicdraw>. Puede descargarse una demostración de la versión *Professional Java* que le servirá para la práctica (para ello, deberá registrarse previamente).