

# Introducción al Portal personalizable de GBIF con extensiones

María Mora  
Administradora del Nodo GBIF  
Costa Rica

# Agenda

- Introducción
- El Portal personalizable de GBIF
  - Arquitectura
  - Tecnologías utilizadas
- Extensiones al Portal GBIF (GBIF.ES, IABIN)
- Otros portales de biodiversidad

# Introducción

- Misión de GBIF ->Hacer que los datos primarios de la biodiversidad del mundo estén disponibles de forma libre y universal por medio de desarrollar una red de bases de datos con acceso por medio de Internet.
- La red debe ser flexible para solventar las necesidades de los proveedores de datos y de los usuarios.

Fuente: [www.gbif.org](http://www.gbif.org)

# Introducción

- Portal personalizable de GBIF
  - Un desarrollo con el soporte de una institución con amplia experiencia en el manejo de información biológica.
  - Gratis y open source.
  - En constante mejora por parte de los desarrolladores.

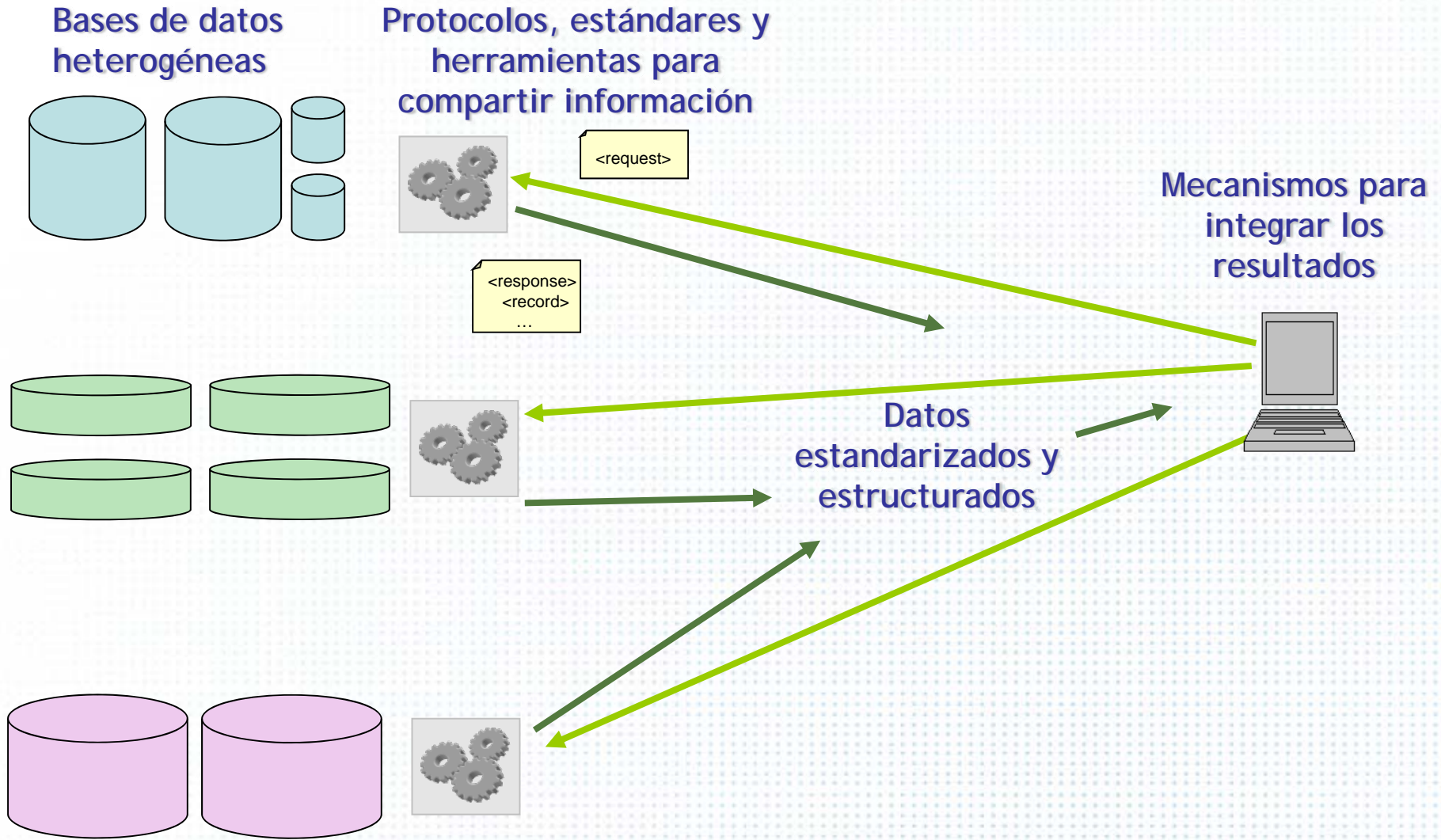
# Introducción

- Tipos de datos
  - Datos taxonómicos:
    - » Nombres científicos y sinónimos
    - » Taxonomía superior
  - Registros de especímenes de colecciones biológicas
  - Registros de observaciones
  - Metadatos de instituciones proveedoras de datos y sus recursos de datos.

# Introducción

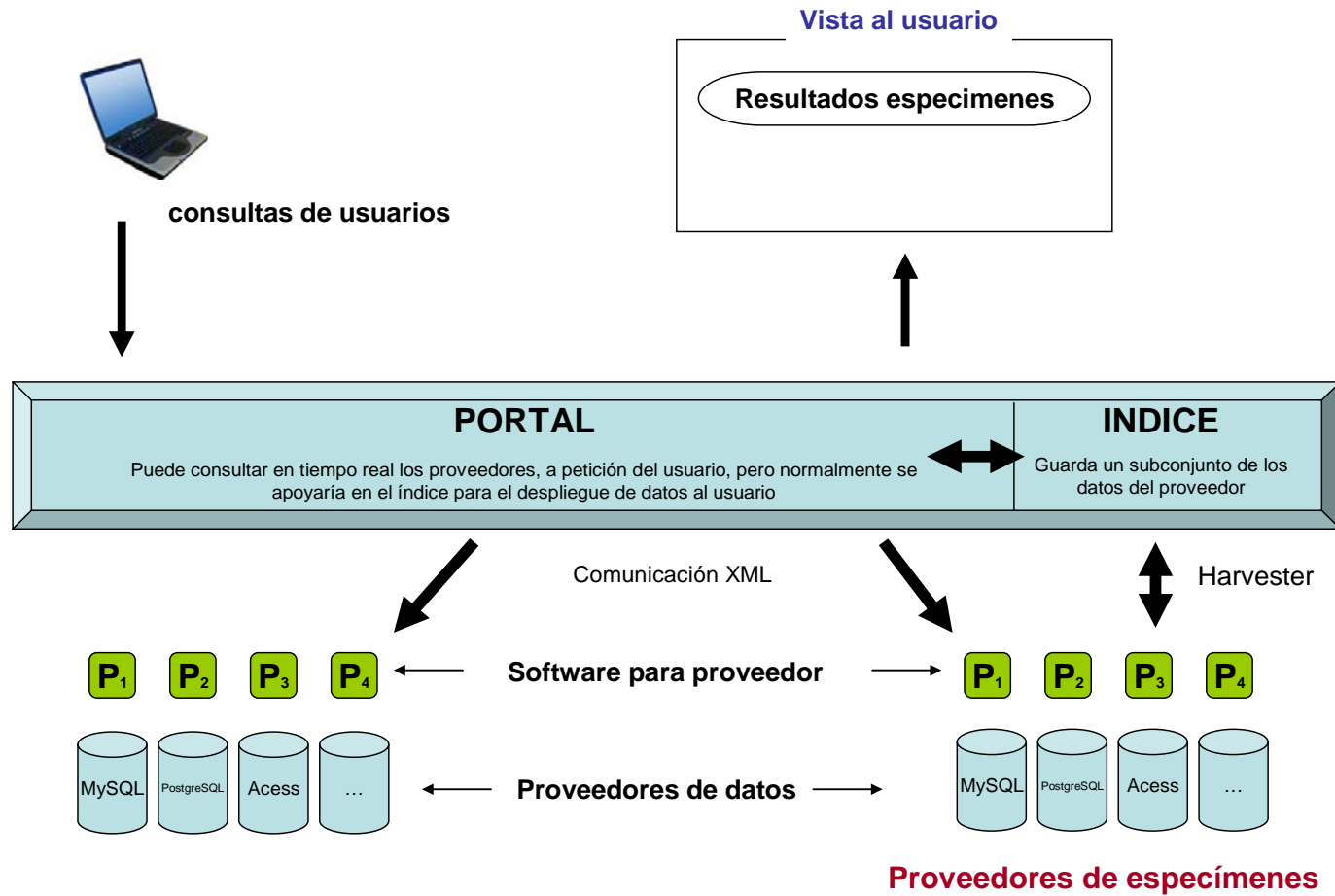
- Tipo de funcionalidad
  - Integración de datos de múltiples bases de datos heterogéneas
  - Búsqueda por medio de algunos conceptos del Darwin Core como: país, proveedor, recurso, nombre científico, entre otros campos.
  - Generación de mapas de distribución
  - Soporte para Darwin Core, ABCD
  - Manejo de protocolos de comunicación con proveedores de datos: DiGIR, BioCASE, TAPIR, IPT

# Arquitectura para la integración



Fuente: [www.gbif.org](http://www.gbif.org)

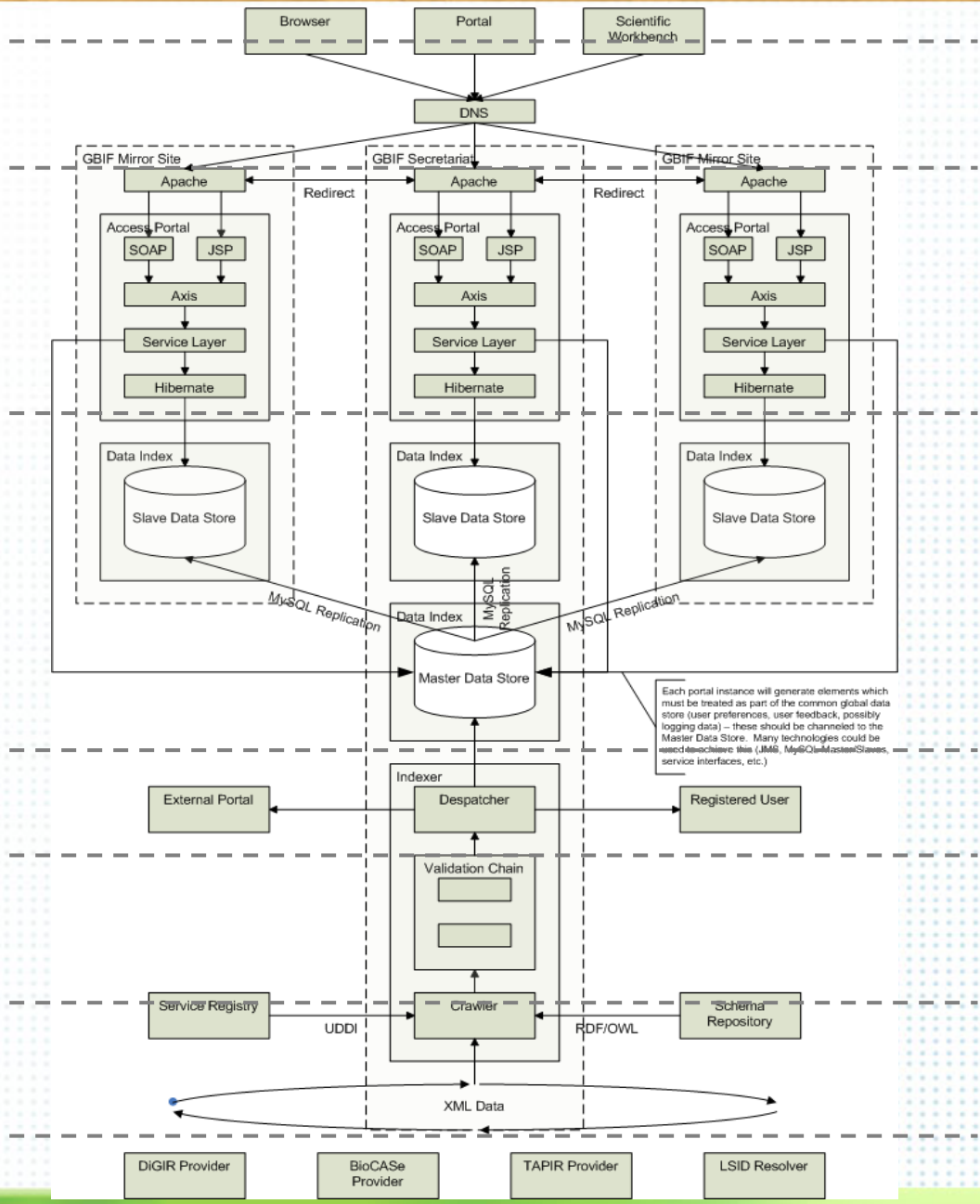
# Arquitectura General





# Arquitectura

# Arquitectura del Portal



**Cientes**

**Espejos**

**Capa web**

**Sincronización de repositorios**

**Distribución de datos**

**Interpretación y validación**

**Recuperación de datos**

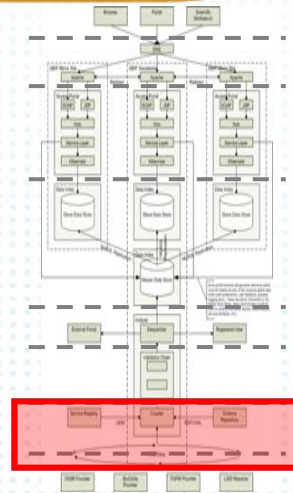
**Recursos de datos**

Fuente: [www.gbif.org](http://www.gbif.org)

# Recursos de datos

Flexibilidad para indexar diferentes tipos de recursos de datos:

- **Registros de especímenes y observaciones**
  - DiGIR con Darwin Core (v. 1.2, v. 1.4, MaNIS, OBIS)
  - BioCASE con ABCD (v. 1.20, v. 1.48, v. 2.06)
  - TAPIR con Darwin Core v. 2.0
  - TAPIR con ABCD v. 2.06
- **Nomenclatura**
  - Información taxonómica con Taxon Concept Schema (TCS)



Fuente: [www.gbif.org](http://www.gbif.org)

# Registro de servicios disponibles

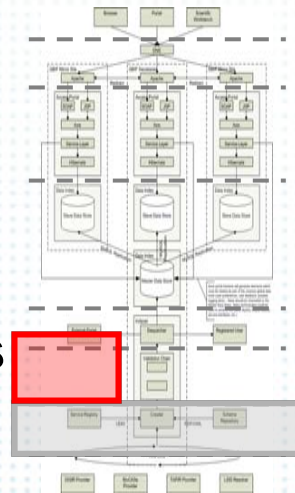
Metadato de los proveedores de datos y sus recursos

- **Mejoras en la descripción y organización de los recursos**

- Categorización de registros por
  - Taxones superiores
  - País proveedor de los registros
  - Países de los cuales provienen los registros
  - Tipo de registro (espécimen, observación, etc)

- **Interfaz de acceso a proveedores de datos**

- Acceso controlado
- Configuración de los períodos de indexación de los recursos
- Revisión del estado de la indexación de cada recurso
- Revisión de informe de indexación y errores detectados durante el proceso
- Revisión de reporte de acceso a los datos



Fuente: [www.gbif.org](http://www.gbif.org)

# Recuperación de datos

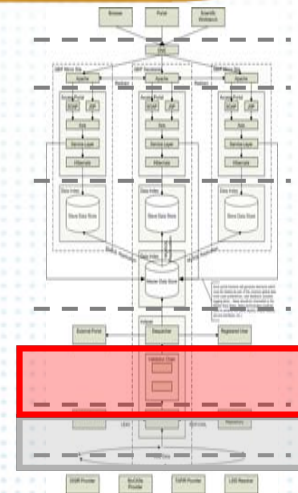
## Búsqueda de datos actualizados y recuperación de registros para ser indexados

- Programación de ejecución

- Selección automática de los recursos a ser indexados
- Indexación multi-nivel (refrescamiento de metadatos, actualización del índice, indexar todo)
- Control del Proveedor de datos de cronograma de actualización de sus recursos

- Enfoque guiado por estrategia

- Usar la experiencia pasada (requests) para la indexación de los recursos para identificar las capacidades del software proveedor.
- Identificar la estrategia para encontrar los registros a indexar (todos, los nuevos, los actualizados, los que previamente fueron inaccesibles o inválidos, etc)
- Desarrollar un mapa del recurso para monitorear el progreso en la indexación, control de registros inaccesibles o inválidos y permitir que la indexación sea detenida temporalmente.
- Recuperar los registros de los recursos de datos de acuerdo al mapa y someterlos a la cadena de validación.



Fuente: [www.gbif.org](http://www.gbif.org)

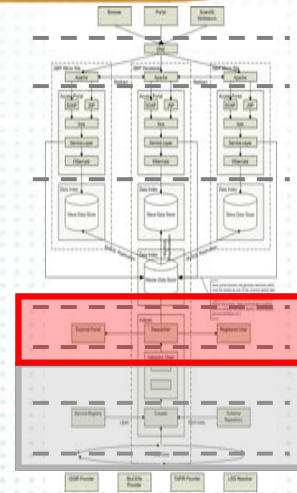
# Interpretación y validación

- **Modelo de flujo de trabajo configurables**

- Manejo de diferentes clases y diferentes formatos de datos
- Facilidad para adicional pasos adicionales y ramas cuando se necesite
- Manejo de taxonomía de virus

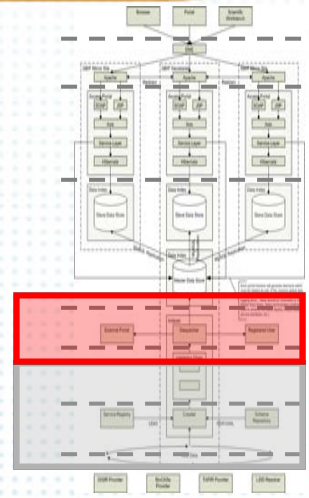
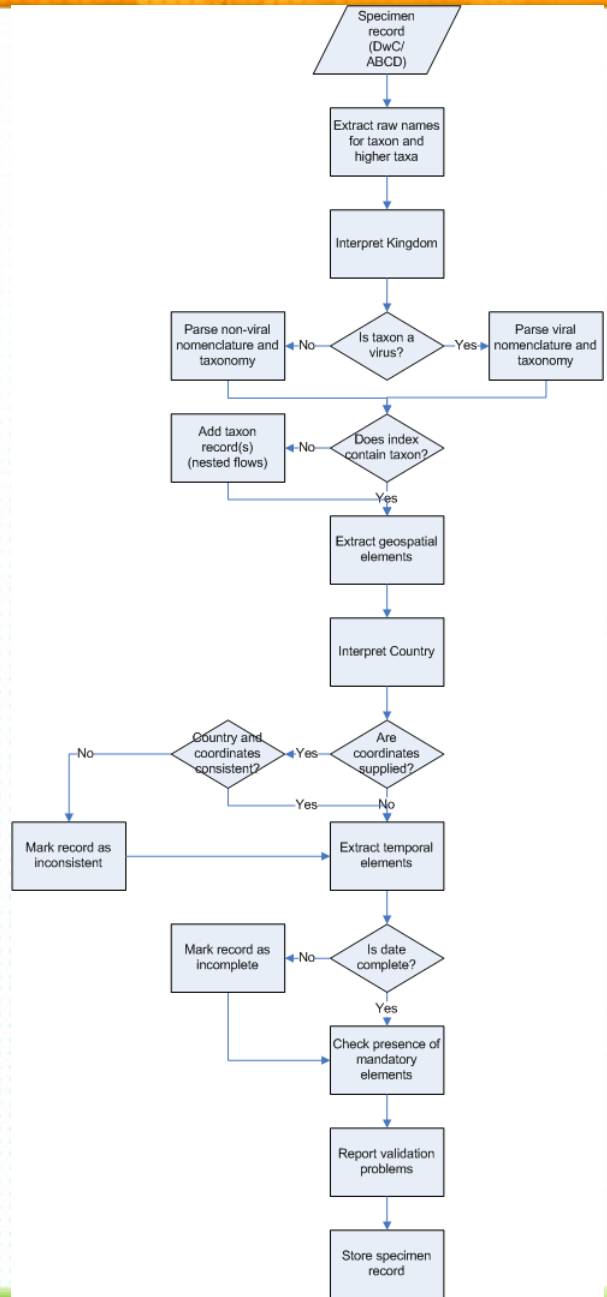
- **Validación de contenido**

- Agregar notas a registros con metadatos para asistir a proveedores y usuarios
- Falta de datos o datos erróneos en campos clave, por ejemplo "basis of record" y scientific name
- Valores incorrectos o contradictorios, por ejemplo latitud y longitud fuera del rango del país.
- Taxones fuera de rango del recurso, ejemplo hongos incluidos en un recurso identificado como proveedor de datos de plantas.
- Generación de informes de errores
- Identificar las fortalezas de los recursos de datos, por ejemplo, datos totalmente georeferenciados.



Fuente: [www.gbif.org](http://www.gbif.org)

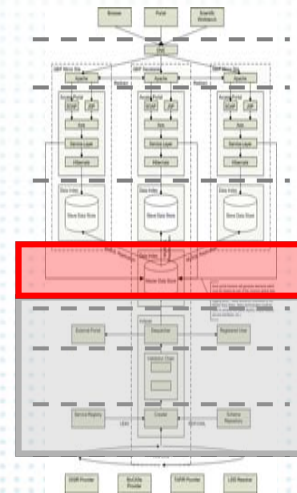
# Interpretación y validación



Fuente: [www.gbif.org](http://www.gbif.org)

# Distribución de datos

- Manejo de los objetos luego del proceso de validación
  - Decisión basaa en las propiedades asignadas a los objetos en el proceso de validación
  -
- Almacenamiento de los objetos en la base de datos del índice



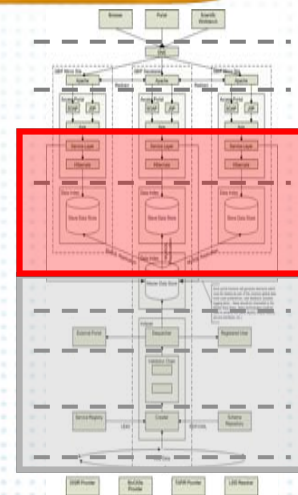
Fuente: [www.gbif.org](http://www.gbif.org)



# Sincronización de los repositorios de datos

Base de datos maestra con espejos de solo lectura

- **Base de datos maestra manejada por el distribuidor de datos**
  - Agregar, modificar o marcar para borrado los registros de acuerdo a lo encontrado por el proceso de recuperación de datos
  - A todos los registros en la base de datos se les asigna un LSIDs interno para mejorar el tiempo de recuperación de los registros.
- **Copias del portal usando bases de datos de solo lectura**
  - Separación de cargas generadas por el proceso de indexación y solicitudes de usuarios
- **Manejo centralizado de estadísticas de uso de los datos.**
  - Recuperación de datos y generación de informes centralizados a partir de todos los datos de uso de la información generados por el portal central y los espejos

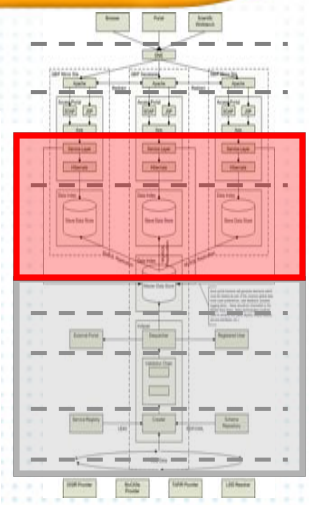
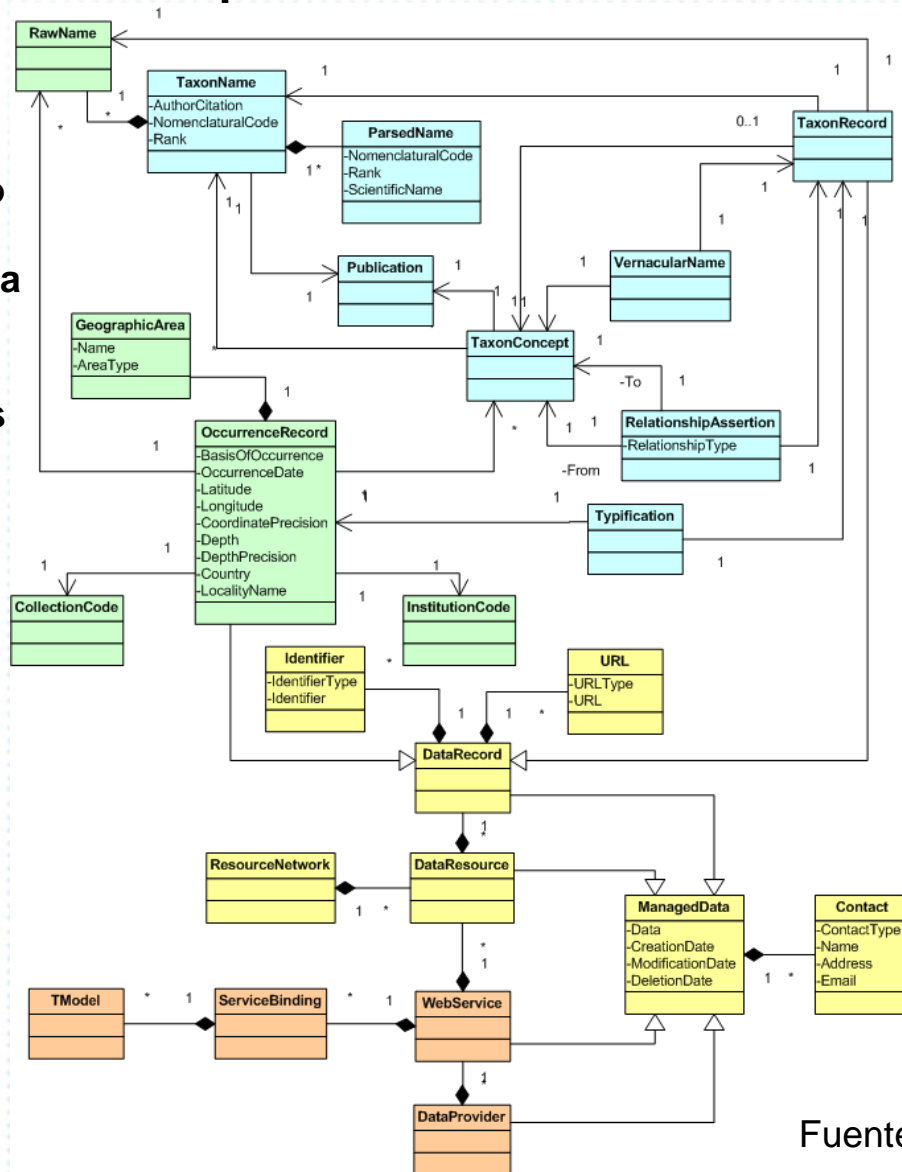


Fuente: [www.gbif.org](http://www.gbif.org)

# Sincronización de los repositorios de datos

## Modelo lógico

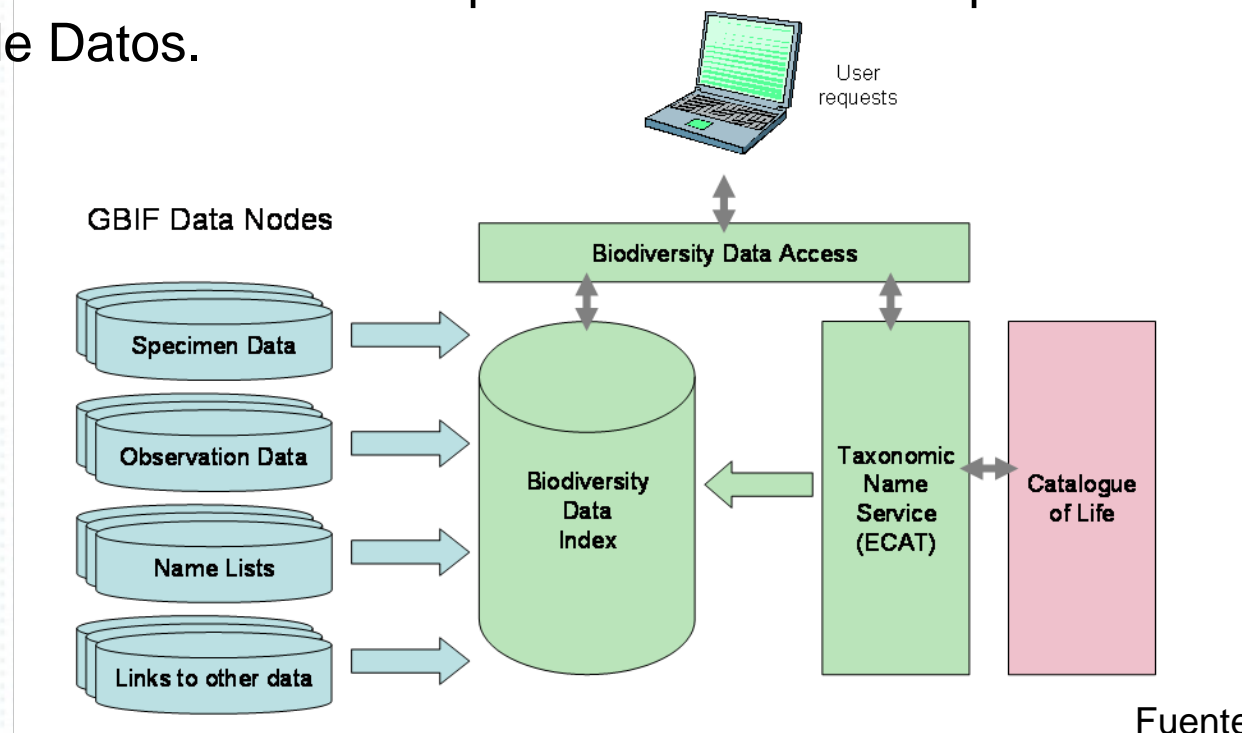
- Naranja: Metadato del registro de proveedores y recursos.
- Datos primarios: nomenclatura y registros taxonómicos
- Verde: Datos primarios de especímenes y observaciones
- Amarillo: metadatos



Fuente: [www.gbif.org](http://www.gbif.org)

# Índice central del Portal GBIF

Una consideración que ha implementado GBIF en la arquitectura actual ha sido la generación de un índice como componente central de la red, lo que asegura que los registros de datos relevantes pueden ser encontrados rápidamente sin tener que visitar cada Nodo de Datos.



Fuente: [www.gbif.org](http://www.gbif.org)

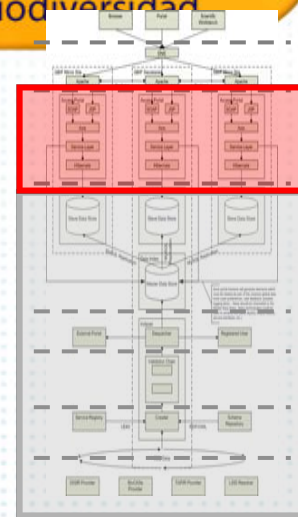
# Índice del Portal

- Metadatos de especímenes/observaciones
- Información taxonómica asociada a un taxón
- Metadatos de los proveedores de datos
- Bitácora de la actividad de los usuarios en el portal

# Capa web

Lógica de aplicación para acceder a los recursos de datos

- **Capa Java para acceso a los datos**
  - Interfaz que soporte todas las solicitudes hecha por la interfaz del portal y servicios web.
- **Uso del framework Model-View-Controller (MVC)**
- **Interfaces HTML configurables**
  - Sistema multilenguaje y formato (layout) configurable
- **Web services**



Fuente: [www.gbif.org](http://www.gbif.org)

See: <http://wiki.gbif.org/dadiwiki/wikka.php?wakka=HTMLUserPortal>

See: <http://wiki.gbif.org/dadiwiki/wikka.php?wakka=XMLDataServices>

# Cientes

## Soporte a web browsers y aplicaciones cliente

- **Interfaz HTML**

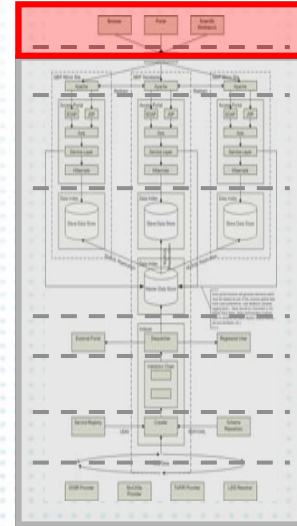
- Capacidad de realizar navegación y búsqueda de datos
- Poder descargar datos delimitados por separadores o en XML
- Implementar capacidad de hacer retroalimentación a los proveedores de datos.
- Opciones de personalización

- **Interfaz del portal expuesta como web services**

- Acceso por medio de WSDL y SOAP a todas las funciones de la interfaz de usuario.

- **Servicios web adicionales para soportar estándares**

- TAPIR con Darwin Core
- TAPIR con ABCD
- Taxon API para Taxon Concept Schema



Fuente: [www.gbif.org](http://www.gbif.org)

# Extensiones al Portal GBIF (GBIF.ES, IABIN, Fundecooperación)

- Incorporar información de fichas de especies:
  - Desarrollo del proceso de indexación
  - Integración de la información taxonómica proveniente de los registros de especies.
  - Búsqueda de fichas de especies.
  - Otra funcionalidad asociada a registros de especies
  - Navegación por medio de nombres comunes.

# Tecnología utilizada



# Spring

- Framework de código abierto, cuyo objetivo es el de hacer más fácil la creación de aplicaciones web basadas en Java y de promover buenas prácticas de programación.

# Hibernate

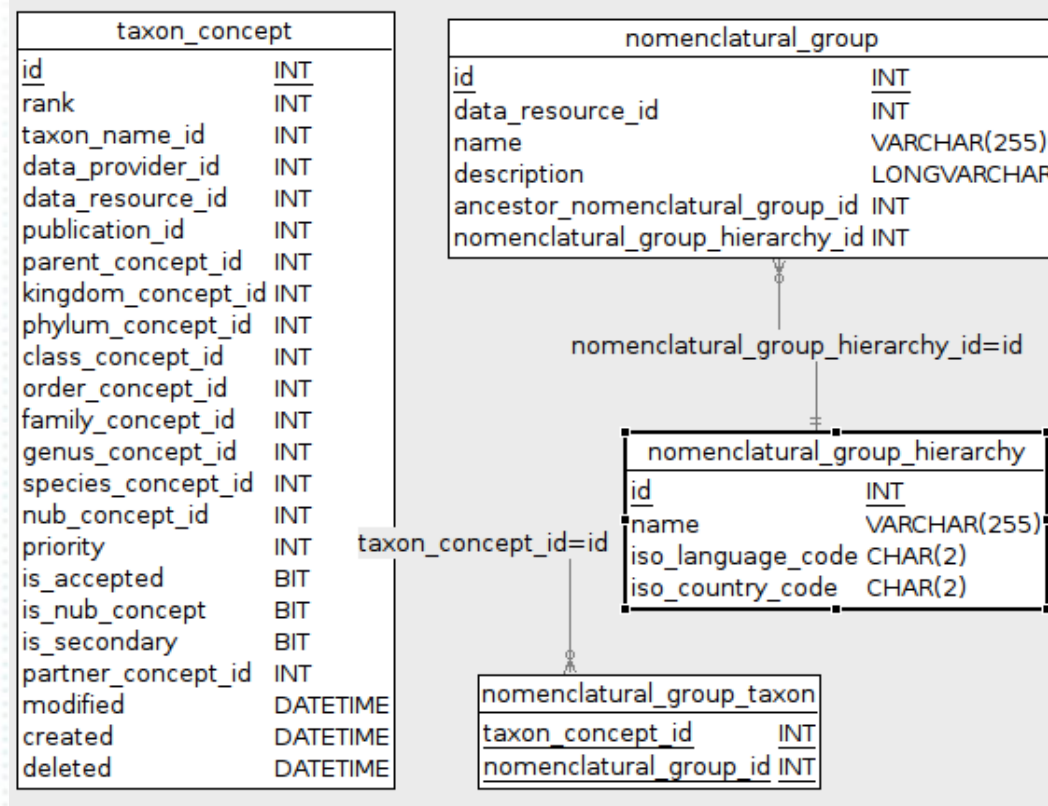
- El código en que está escrito el portal es orientado a objetos y la BD es relacional.
- Esto genera una “incompatibilidad” de paradigmas, que se resuelve usando herramientas para mapeo objetos/relacional (O/R).

# Hibernate

- Hibernate es una implementación de un O/R.
- De este al programar se trabaja como si se tuviera una BD orientada a Objetos.
- Además se obtienen otros beneficios como bajo acoplamiento a la BD.

# Hibernate: Mapeos

- En el modelo relacional:



# Hibernate: Mapeos

- Mapeo Trivial

```
public class NomenclaturalGroupHierarchy extends BaseObject {  
  
    protected Long id;  
    protected String name;  
    protected String isoLanguageCode;  
    protected String isoCountryCode;
```

```
4<hibernate-mapping>  
5  <class table="nomenclatural_group_hierarchy"  
6  name="org.gbif.portal.model.nomenclaturalgroup.NomenclaturalGroupHierarchy">  
7    <id unsaved-value="null" name="id" column="id">  
8      <generator class="native"/>  
9    </id>  
10   <property name="name" column="name" not-null="true"/>  
11   <property name="isoLanguageCode" column="iso_language_code" not-null="true"/>  
12   <property name="isoCountryCode" column="iso_country_code" not-null="true"/>  
13 </class>  
14</hibernate-mapping>
```

# Hibernate: Mapeos

- Mapeo Muchos a Uno

```
28 public class NomenclaturalGroup extends BaseObject {
29
30     protected DataSource dataSource;
31     protected Long dataSourceId;
32     protected String name;
33     protected String description;
34     protected NomenclaturalGroup ancestor;
35     protected Long ancestorId;
36     protected NomenclaturalGroupHierarchy groupHierarchy;
37     protected Set<TaxonConcept> taxonConcepts;
38 }
```

# Hibernate: Mapeos

- Mapeo Muchos a Uno

```
4<hibernate-mapping>
5  <class table="nomenclatural_group"
6    name="org.gbif.portal.model.nomenclaturalgroup.NomenclaturalGroup">
7    <id unsaved-value="null" name="id" column="id">
8      <generator class="native" />
9    </id>
10   <many-to-one column="data_resource_id" cascade="save-update"
11     not-null="false" name="dataResource" />
12   <property name="name" column="name" not-null="true" />
13   <property name="description" column="description"
14     not-null="true" />
15   <many-to-one column="ancestor_nomenclatural_group_id"
16     cascade="save-update" not-null="false" name="ancestor" />
17   <property name="ancestorId"
18     column="ancestor_nomenclatural_group_id" not-null="true"
19     insert="false" update="false" />
20   <many-to-one column="nomenclatural_group_hierarchy_id"
21     cascade="save-update" not-null="false" name="groupHierarchy" />
22   <set lazy="true" name="taxonConcepts"
23     table="nomenclatural_group_taxon">
24     <key column="nomenclatural_group_id" />
25     <many-to-many
26       class="org.gbif.portal.model.taxonomy.TaxonConcept"
27       column="taxon_concept_id" lazy="proxy" />
28   </set>
29 </class>
30</hibernate-mapping>
```

# Hibernate: Mapeos

- Mapeo Muchos a Muchos

```
28 public class NomenclaturalGroup extends BaseObject {
29
30     protected DataSource dataSource;
31     protected Long dataSourceId;
32     protected String name;
33     protected String description;
34     protected NomenclaturalGroup ancestor;
35     protected Long ancestorId;
36     protected NomenclaturalGroupHierarchy groupHierarchy;
37     protected Set<TaxonConcept> taxonConcepts;
38 }
```



# Hibernate: Mapeos

- Mapeo Muchos a Muchos

```

4<hibernate-mapping>
5  <class table="nomenclatural_group"
6    name="org.gbif.portal.model.nomenclaturalgroup.NomenclaturalGroup">
7    <id unsaved-value="null" name="id" column="id">
8      <generator class="native" />
9    </id>
10   <many-to-one column="data_resource_id" cascade="save-update"
11     not-null="false" name="dataResource" />
12   <property name="name" column="name" not-null="true" />
13   <property name="description" column="description"
14     not-null="true" />
15   <many-to-one column="ancestor_nomenclatural_group_id"
16     cascade="save-update" not-null="false" name="ancestor" />
17   <property name="ancestorId"
18     column="ancestor_nomenclatural_group_id" not-null="true"
19     insert="false" update="false" />
20   <many-to-one column="nomenclatural_group_hierarchy_id"
21     cascade="save update" not null="false" name="groupHierarchy" />
22   <set lazy="true" name="taxonConcepts"
23     table="nomenclatural_group_taxon">
24     <key column="nomenclatural_group_id" />
25     <many-to-many
26       class="org.gbif.portal.model.taxonomy.TaxonConcept"
27       column="taxon_concept_id" lazy="proxy" />
28   </set>
29 </class>
30</hibernate-mapping>

```

# Hibernate: Consultas

- Hibernate define su propio lenguaje orientado a objetos para hacer consultas, HSQL.
- Para comparar con el SQL se hará como ejemplo una consulta para obtener el nombre de la jerarquía nomenclatural al que pertenece el grupo nomenclatural 'lagartijas'.

# Hibernate: Consultas

- SQL

```
mysql> select ngh.name  
-> from nomenclatural_group_hierarchy as ngh,  
->      nomenclatural_group as ng  
-> where ng.nomenclatural_group_hierarchy_id = ngh.id  
-> and ng.name = 'lagartijas';
```

```
+-----+  
| name          |  
+-----+  
| Nombres Comunes |  
+-----+  
1 row in set (0.00 sec)
```

# Hibernate: Consultas

- HSQL

```
select ng.groupHierarchy.name  
from NomenclaturalGroup as ng  
where ng.name = 'lagartijas'
```

Hibernate Query Result

0

Nombres Comunes

# MAVEN

- Esta herramienta facilita el manejo de los proyectos y sus dependencias
- Consulta repositorio central en búsqueda de dependencias
- Ciclo de vida de Maven es:
  - Compilar
  - Probar
  - Empaquetar
  - Instalar
  - Deploy

# Portales para acceso a información de biodiversidad

- INBio
  - <http://atta.inbio.ac.cr/>
- OTS – La Selva
  - <http://sura.ots.ac.cr/local/florula3/index.htm>
- ORNIS
  - <http://olla.berkeley.edu/ornisnet/>
- MaNIS
  - <http://manisnet.org/>
- GBIF
  - <http://www.gbif.org/>
- IABIN
  - <http://www.iabin.net/>

Muchas Gracias